Slackware Linux Essentials
O Guia Oficial do Slackware Linux



Projeto de tradução do Slackbook para Português do Brasil
Slackbookptbr - http://slackbookptbr.sf.net
Versão: 1.0.2

Slackware Linux Essentials
Segunda Edição
Publicação: Maio 2005
isbn: 1-57176-338-4

Slackware Linux Essentials, Second Edition

Published by Slackware Linux, Inc., 1164 Claremont Drive, Brentwood, CA 94513

Lead Author, Second Edition: Alan Hicks.
Editors, Second Edition: Murray Stokely and FuKang Chen.
Authors, First Edition: Chris Lumens, David Cantrell, and Logan Johnson.

Print History:
June, 2000 First Edition
May, 2005 Second Edition

Sumário

Prefácio

Público Alvo

O sistema operacional Slackware Linux é uma poderosa plataforma baseada em computadores *i386*. Ele foi projetado para ser estável, seguro e funcional como servidor, e também uma poderosa estação de trabalho.

Este livro foi feito para que você comece com o sistema operacional Slackware Linux. Isto não siginifica que iremos abordar cada detalhezinho da distribuição, mas mostraremos rapidamente do que ela é capaz dando o conhecimento para trabalhar com o básico do sistema.

Como você ganhará experiência com o Slackware, esperamos que você busque este livro para ser uma referência acessível. Nós também esperamos que o empreste para os seus amigos quando eles vierem perguntar sobre o sistema operacional Linux que você está usando.

Mesmo que esse livro não vá fazer você pular de emoção, nós certamente tentamos fazê-lo o mais interessante possível. Com alguma sorte dará um filme. Claro, nós esperamos também que você possa aprender com ele e o ache útil.

E agora, comecemos com o show.

Mudanças da Primeira Edição

Esta é a segunda edição, resultado de vários anos de trabalho duro dos membros dedicados do Projeto de Documentação do Slackware. A seguir estão as mudanças de maior valor nesta nova edição:

- Capítulo 3, O capítulo Instalação foi modificado com novas fotos da tela do instalador, alterações decorrentes no conjunto de discos, e a instalação a partir do CD.
- Capítulo 4, O capítulo Configuração do Sistema, foi atualizado com novas informações sobre o Kernel 2.6.x do Linux.
- Capítulo 5, O capítulo sobre Configuração de Rede foi expandido, com explicações adicionais do Samba, NFS, e DHCP. A seção "redes sem fio"também foi adicionada. Este capítulo agora está repleto de grandes mudanças, sobre como instalar redes seguras no Slackware.
- Capítulo 6, O capítulo sobre Sistema de janela X, foi substancialmente re-escrito para sistemas baseados no Xorg. Este capítulo agora cobre também o gerenciador gráfico de login xdm.
- Capítulo 13, O capítulo "comandos de rede básicos", foi realçado com informações sobre utilitários de redes adicionais.
- Capítulo 14, Segurança, é um novo capítulo desta edição. Ele explica como manter um sistema Slackware Linux seguro.
- Capítulo 17, Emacs, é um novo capítulo desta edição. Ele descreve como usar o Emacs, um editor poderoso para Unix.
- Capítulo 18, O capítulo "Gerênciamento de Pacotes", foi atualizado com informações sobre scripts do SlackBuild.
- Existem ainda muitas outras mudanças, de menor e maior importância, que refletem as mudanças de como o Slackware amadureceu.

Organização deste Livro

Capítulo 1, Introdução

- Fornece um material introdutório sobre o Slackware Linux, e os Movimentos de Software Livre e Código Aberto.

Capítulo 2, Ajuda

- Descreve os recursos de ajuda disponíveis no sistema Slackware Linux e na internet.

Capítulo 3, Instalação

- Descreve o processo de instalação passo-a-passo com fotos de tela para lhe mostrar como fazê-la por meio de ilustrações.

Capítulo 4, Configuração do Sistema

- Descreve os importantes arquivos de configuração e fala sobre a recompilação do kernel.

Capítulo 5, Configuração da Rede

- Descreve como conectar uma máquina Slackware Linux à rede. Fala sobre TCP/IP, PPP/dial-up, rede wireless, e muito mais.

Capítulo 6, O Sistema de Janelas X

- Descreve como instalar o Sistema gráfico de Janelas X no Slackware.

Capítulo 7, Inicialização

- Descreve o processo pelo qual o computador inicia no Slackware Linux. Fala também sobre a dupla inicialização com o sistema operacional Microsoft Windows.

Capítulo 8, O Shell

- Descreve a poderosa interface de linha de comando do linux.

Capítulo 9, Estrutura do Sistema de Arquivos

- Descreve a estrutura do sistema de arquivos, incluindo dono de arquivos, permissões, e ligações simbólicas(links).

Capítulo 10, Manipulando Arquivos e Diretórios

- Descreve os comandos usados para manipular arquivos e diretórios através da interface de linha de comando.

Capítulo 11, Controle de Processos

- Descreve os poderosos comandos de gerênciamento de processos do Slackware, usados para gerenciar aplicativos sendo executados paralelamente.

Capítulo 12, Fundamentos para Administração do Sistema

- Descreve a administração básica de tarefas do sistema como adicionar e remover usuários, desligar o sistema corretamente e muito mais.

Capítulo 13, Comandos Básicos de Rede

- Descreve a coleção de clientes de rede incluídos no Slackware.

Capítulo 14, Segurança

- Descreve várias ferramentas diferentes disponíveis para ajudá-lo a manter seguro seu sistema Slackware, incluindo *iptables* e *tcpwrappers*.

Capítulo 15, Arquivos Compactados

- Descreve compressões diferentes e utilidades de arquivos disponíveis no Linux.

Capítulo 16, vi

- Descreve o poderoso editor de texto *vi*.

Capítulo 17, Emacs

- Descreve o poderoso editor de texto *Emacs*.

Capítulo 18, Gerenciamento de Pacotes do Slackware

- Descreve os utilitários de empacotamento do Slackware e o processo usado para criar pacotes padronizados e tagfiles.

Capítulo 19, ZipSlack

- Descreve a versão ZipSlack do Linux que pode ser usada a partir do Windows sem precisar instalá-la.

Appendix A, Licença Pública Geral GNU

- Descreve os termos de licença sob os quais o Linux Slackware e este livro podem ser copiados e distribuídos.

Convenções usadas neste livro

Para fornecer uma consistente e fácil leitura do texto, diversas convenções foram seguidas durante todo o livro.

Convenções Tipográficas

- *Itálico*

  \- Fontes em *itálico* são usada para comandos, texto enfatizado, e os termos técnicos de primeiro uso.

- **Negrito**

  – Uma fonte em **negrito** é usada para entrada de dados do usuário nos exemplos.

Entrada do Usuário

Teclas são mostradas em **negrito** para se destacarem de outros textos. Combinações de teclas que possuem certo objetivo por serem digitadas simultaneamente, são mostradas com '+' entre as teclas, como:

Ctrl + **Alt** + **Del**

Desse modo o usuário deve digitar as teclas **Ctrl**, **Alt**, e **Del** ao mesmo tempo.

Teclas que possuem certo objetivo por serem digitadas em sequência que irão ser separadas por vírgulas, por exemplo:

Ctrl+**X**, **Ctrl**+**S**

Significaria que é esperado que o usuário digite as teclas **Ctrl** e **X** simultaneamente e então digite as teclas **Ctrl** e **S** simultaneamente.

Exemplos

Exemplos começando com E:/> indica um comando MS-DOS®. A menos que seja conhecido de outra forma, estes comandos podem ser executados de uma janela de "Prompt de Comandos"do ambiente moderno do Microsoft Windows.

D:> rawrite a: bare.i

Exemplos começando com &prompt.root; indica que o comando que deve ser usado pelo super-usuário no Slackware. Você pode iniciar uma sessão como *root* para digitar o comando, ou iniciar com uma conta normal e usar &man.su.1; para ganhar privilégio de super-usuário.

```
# dd if=bare.i of=/dev/fd0
```

Exemplos começando com &prompt.user; indicam que o comando pode ser usado em uma conta de usuário normal. A menos que seja conhecido de outra forma, a sintaxe C-shell é usada para ajustar as variáveis de ambiente e outros comandos de shell.

```
$ top
```

Agradecimentos

Este projeto é o acumulo de meses de trabalho de muitas pessoas dedicadas. Não seria possível para mim produzir este trabalho do zero. Muitas pessoas merecem nossos agradecimentos por seus atos de ajuda: Keith Keller por seu trabalho em rede wireless, Joost Kremers por seu maravilhoso trabalho escrito sem colaboradores da seção Emacs, Simon Williams por seu capítulo sobre segurança, Jurgen Phillippaerts pelos comandos básicos de rede, Cibao Cu Ali G Colibri pela inspiração e o ponta-pé inicial. Sem contar outros que nos enviaram sugestões e reparos. Uma lista incompleta inclui: Jacob Anhoej, John Yast, Sally Welch, Morgan Landry, e Charlie Law. Tenho também de agradecer Keith Keller por hospedar a lista de emails desse projeto, bem como Carl Inglis pela hospedagem inicial do site. Por último mas não menos importante, tenho que agradecer a Patrick J. Volkerding pelo Slackware, e David Cantrell, Logan Johnson, e Chris Lumens pelo Fundamentos do Slackware Linux 1ª Ediçao. Sem sua estrutura inicial, nada disso teria acontecido. Muitos outros contribuiram

com pequenas e grandes partes desse projeto e não foram listadas. Espero que me perdoem por minha memória fraca.

Alan Hicks, Maio 2005

Capítulo 1

Uma Introdução ao Slackware Linux

1.1 O que é o Linux ?

Linus Torvalds criou o Linux, o núcleo do sistema operacional, como um projeto pessoal em 1991. Ele iniciou o projeto porque procurava rodar um sistema operacional baseado em Unix mas não possuia muito dinheiro. além disso, Ele procurava aprender sobre a arquitetura do processador 386. O Linux é liberado livre de cobrança para o público assim qualquer pessoa pode ajudar estudando-o e fazendo melhorias sobre a licença pública geral. (veja Código Aberto e Software Livre e Apêndice A para uma explicação da licença.) Hoje o linux cresceu como um dos principais jogadores no mercado de sistema operacional. Ele foi portado para rodar em uma variedade de arquiteturas, incluindo HP/Compaq's Alpha, Sun's SPARC e UltraSPARC, e Motorola's PowerPC chips (através Apple Macintosh e computadores IBM RS/6000.) centenas, se não milhares, de programadores em todo o mundo desenvolve o Linux. Ele roda programas como Sendmail, Apache, e BIND, que são softwares bastante populares usado em servidores de internet. É importante lembrar que o termo *"Linux"* realmente se refere ao kernel - o núcleo do sistema operacional. Este núcleo é responsável por controlar o processador do seu computador, memória, discos rígidos e periféricos. Aquilo tudo é realmente o Linux: Controla as operações do seu computador e certifica-se de que todos os seus programas se comportam. Existem várias companhias e indivíduos que empacotam o kernel e vários programas juntos para fazer um sistema operacional. Nós chamamos cada pacote de uma distribuição Linux.

1.2 Uma palavrinha sobre GNU

O projeto do Kernel Linux começou com um esforço pessoal de Linus Torvalds em 1991, mas como Isaac Newton disse uma vez,*"Se consegui enxergar mais longe, é porque estava apoiado sobre ombros de gigantes."* Quando Linus Torvalds criou o Kernel a Free Software Foundation tinha estabelecido já a idéia do software colaborativo. Intitularam seu esforço GNU, um acrônimo recursivo que significa simplesmente *GNU's Not Unix*. O Software da GNU funciona sobre o kernel do Linux desde o 1º dia. O compilador *gcc* foi usado usado para compilar o kernel. Hoje muitas ferramentas da GNU, do *gcc* ao *gnutar* estão ainda na base de cada distribuição principal do Linux. Por esta razão muitos dos membros da Free Software Foundation sugerem fervorosamente que seu trabalho deve ser dado o mesmo crédito que o kernel Linux e ainda que todas as distribuições de Linux devem se referir a ele como distribuições de GNU/Linux.

O prefixo "GNU"em "GNU/Linux"é o tema de inúmeras flamewars; só a guerra santa entre Emacs e vi gera mais brigas. O objetivo deste livro não é alimentar as chamas da discussão, e sim esclarecer a terminologia para os neófito. Quando vemos "GNU/Linux"isso quer dizer uma distribuição Linux; quando vemos "Linux"o termo pode estar se referindo tanto ao kernel quanto à distribuição inteira, o que pode ser um pouco confuso; e o termo "GNU/Linux"muitas vezes não é usado por ser longo demais.

1.3 O que é o Slackware?

O Slackware, criado por Patrick Volkerding no fim de 1992, e inicialmente lançado para o mundo em 17 de Julho de 1993, foi a primeira distribuição Linux a alcançar um uso bastante difundido.

Volkerding iniciou o aprendizado no linux quando precisou de um interpretador de LISP para um projeto. Uma das poucas distribuições disponíveis naquele tempo era a SLS Linux de Soft Landing Systems. Volkerding usou o SLS Linux, corringindo os bugs que encontrava.

Eventualmente, decidiu juntar todos estes bugfixes em sua própria distribuição privada que ele e seus amigos poderiam usar. Esta distribuição privada ganhou rapidamente a popularidade, assim que Volkerding decidiu nomeá-lo para Slackware e distribui-lo publicamente. Ao longo, Patrick adicionou coisas novas ao Slackware; um programa de instalação amigável para o usuário e baseado em um sistema de menu, também com o conceito de gerênciamento de pacote, na qual permite que os usuários facilmente adicionem, removam ou atualizem um programa ou o seu sistema inteiro.

Há muitas razões porque o Slackware é a distribuição Linux mais antiga em atividade. Não tenta imitar o Windows, tenta ser o mais similar ao Unix possível. Ele não esconde seus processos atrás de uma interface gráfica de apontar e clicar; ao invés disso ele põe os usuários no controle e mostra para eles exatamente o que está acontecendo. Seu desenvolvimento não é acelerado artificialmente para cumprir prazos - cada versão só é lançada quando realmente está pronta.

O Slackware é para as pessoas que gostam de aprender e fazer com o sistema exatamente o que quer. O que permite que o Slackware será usado por vários anos é a sua simplicidade e estabilidade. O Slackware atualmente possui uma reputação de um servidor estável e uma rápida estação de trabalho, Você poderá encontrar desktops com Slackware funcionando quase todos com um gerenciador de janelas ou com ambiente de trabalho, ou sem nenhum. Os Servidores Slackware aumentam os negócios, aumentando a capacidade para os usuários do servidor, Os usuários de Slackware são entre todos usuários os mais satisfeitos com o Linux. Naturalmente, nós diríamos isso. :ˆ)

1.4 Código Aberto e Software Livre

Dentro da comunidade Linux, existem dois principais movimentos ideológicos em atividade. O movimento de Software livre (nós começaremos em instantes) que está trabalhando com o objetivo de fazer todo o software livre das restrições da propriedade intelectual. Os seguidores deste movimento acreditam que as restriçoes impedem as melhorias tecnicas e o trabalho contrário ao bem da comunidade. O movimento de código aberto está trabalhando praticamente com os mesmos objetivos, mas tomadas uma aproximação mais pragmática a elas. Os seguidores deste movimento preferem basear seus argumentos nos méritos econômicos e técnicos de fazer o código fonte livremente disponível, melhor que os princípios morais e éticos que dirigem o movimento de Software Livre, No extremo oposto da visão estão os grupos que desejam manter um controle em exesso sobre seus softwares.

O movimento de software livre é dirigido pela Free Software Foundation, uma organização de levantamento de fundos para o projeto GNU. O software livre é mais que uma ideologia. A expressão usada com frequência é *livre como um discurso, mas não como cerveja grátis.* Essencialmente, o software livre é uma tentativa de garantir determinados direitos para os usuários e desenvolvedores. Esta liberdade incluem a liberdade para executar o programa sem uma razão, para estudar e modificar o código fonte, para redistribuir o código fonte e para compartilhar de quaisquer modificações que você fizer. A fim de garantir esta liberdade, a licença pública geral da GNU (GPL) foi criado. A GPL, sendo breve, prover que qualquer um que distribui um programa compilado que seja licenciado sob a GPL deve também fornecer o código fonte, e é livre para fazer modificacões no programa e aquelas modificações feitas também estarão disponíveis na forma de código fonte. Isto garante que uma vez que um programa é *aberto* à comunidade, não pode ser *fechado* exceto pelo consentimento de cada autor de cada parte do código (mesmo as modificações) dentro dele. A maioria dos programas para o Linux estão licenciados sob a GPL.

É importante notar que a GPL não diz qualquer coisa sobre o preço. Tão ímpar como pode soar, você pode cobrar pelo software livre. A parte *livre* está na liberdade que você tem com o código fonte, não no preço que você paga pelo software. (entretanto, uma vez que alguém o vendeu, ou tenha dado um programa compilado licenciado sob a GPL são obrigados a fornecer também o código fonte.)

Outra licença popular é a licença BSD. Em contraste com a GPL, a licença BSD não impõe nenhuma exigência para a liberação do código fonte de um programa. Os Softwares lançado sobre a licença BSD permitem redistribuir em código fonte ou em binário uma forma de providenciar apenas algumas condições. O crédito do autor não pode ser usados como um caráter de propaganda para o programa. Imuniza também o autor da responsabilidade para os danos que podem se levados pelo uso do software. Muito dos softwares incluído no Slackware Linux são licenciados pela BSD.

Na linha de frente do jovem movimento de código aberto, A OSI é uma organização que existe unicamente para avançar a sustentação para o software de código aberto, isto é, o software ter o código aberto disponivel

perfeitamente para ler e executar o programa. Não oferecem uma licença específica, mas em substituição suportam os vários tipos de licenças de códigos abertos.

A ideia atrás da OSI é trazer mais companhias para o código aberto permitindo que escrevam suas próprias licenças de código aberto e tenham aquelas licenças certificadas pela Open Source Initiative. Muitas companhias querem liberar o código fonte, mas não querem usar a GPL. Desde que não podem radicalmente mudar a GPL, são oferecidas a oportunidade de fornecer sua própria licença e de tê-la certificada por esta organização.

Quando a Free Software Foundation e a Open Source Initiative trabalharem para se ajudarem, elas não são a mesma coisa. A Free Software Foundation usa uma licença específica e fornece o software sob esta licença, A Open Source Initiative busca dar suporte para todas a licenças abertas, incluindo uma para a Free Software Foundation. Toda a sustentação aberta das buscas da Open Source Initiative para abrir o código fonte, inluindo a Free Software Foundation. As regiões em que cada uma discute fazendo o código fonte livremente disponível dividem às vezes os dois movimentos, mas o fato de que dois grupos ideologicamente diferentes estão trabalhando para o mesmo objetivo, isso dar crédito aos esforços de cada um.

Capítulo 2

Ajuda

Muitas vezes você precisa de ajuda para um comando específico, para iniciar um programa, ou simplesmente saber como um hardware trabalha. Talvez você queira saber mais opções que possam ser utilizadas em um determinado comando, ou apenas verificar quais opções estão disponíveis para serem utilizadas em um determinado comando. Felizmente, existem várias maneiras de você obter ajuda para o que está procurando. Quando você instala o Slackware você tem a opção de instalar os pacotes da série *F* que inclui FAQs e HOWTOs. Alguns programas também possuem ajuda sobre outras opções, arquivos de configuração e uso.

2.1 Ajuda do Sistema

2.1.1 man

O comando *man* (abreviação de *manual*) é a forma tradicional de documentação online dos sistemas operacionais Unix e Linux. Composto de Linhas formatadas especialmente, as *man page*, são escritas para a grande maioria dos programas e são distribuídas com o próprio software. Executando *man comando* será mostrada a man page para o programa especificado, em nosso exemplo esse seria o comando imaginário *comando*.

Você deve imaginar a grande quantidade de páginas de manual que podem ser rapidamente criadas, tornaria-se confuso e complicado até mesmo para um usuário avançado. Devido a isso, às páginas de manual são divididas em seções numeradas. Este sistema esteve disponível por um longo tempo; você verá bastante comandos, programas e referências de funções de bibliotecas referênciadas pelo número da seção do man.

Por exemplo:

Você pode fazer uma referência ao *man*(1). O número especifica que a *man* é documentada na seção 1 (comandos de usuários); você pode especificar que você quer a documentação da seção 1 do *man* com o comando *man 1 man*. Especificar a seção é útil quando existe manuais em várias seções com o mesmo nome.

Seção	Conteúdo
Seção 1	Comandos de usuários (somente introdução)
Seção 2	hamadas do Sistema
Seção 3	Chamadas de bibliotecas C
Seção 4	Dispositivos (ex., hd, sd)
Seção 5	Formatos de Arquivos e protocolos (ex., wtmp, /etc/passwd, nfs)
Seção 6	Jogos (somente introdução)
Seção 7	Convenções, pacotes macro, etc. (ex., nroff, ascii)
Seção 8	Administração do sistema (somente introdução)

Em adicional ao comando *man*(1), existem os comandos *whatis*(1) e *apropos*(1) que estão disponíveis para você, com a finalidade de encontrar uma informação mais fácil no sistema.

O comando *whatis* dá uma descrição muito breve dos comandos do sistema, uma espécie de referência.

Exemplo:

```
$ whatis whatis
whatis (1) - search the whatis database for complete words
```

O comando *apropos* é usado para procurar a página de manual que contêm uma determinada palavra..

Exemplo:

$ apropos wav

cdda2wav (1) - a sampling utility that dumps CD audio data into wav sound files
netwave_cs (4) - Xircom Creditcard Netwave device driver
oggdec (1) - simple decoder, Ogg Vorbis file to PCM audio file (WAV or RAW)
wavelan (4) - AT&T GIS WaveLAN ISA device driver
wavelan_cs (4) - AT&T GIS WaveLAN PCMCIA device driver
wvlan_cs (4) - Lucent WaveLAN/IEEE 802.11 device driver

Caso queira alguma informação adicional para algum destes comandos, leia a página de manual para maiores detalhes. ;)

2.1.2 O diretório /usr/doc/

A grande maioria dos softwares que instalamos vem com algum tipo de documentação: arquivos README, instruções de uso, arquivos de licença, etc. Geralmente toda documentaçao que vêm com o software é instalada no sistema no diretório */usr/doc/*. Cada programa geralmente instala sua documentação na seguinte ordem:

/usr/doc/$programa-$versão/

Onde $programa é o nome do programa que você está querendo obter informações, e $versão// é (obviamente) a versão do software que está instalado em seu sistema.

Por exemplo, para ler a documentação do comando *man*(1) você digitaria *cd* no:

$ cd */usr/doc/man-$versão/*

Caso ao ler as informações da página de manual você não encontre informações apropriadas o próximo passo é buscar informações no diretório */usr/doc/*.

2.1.3 HOWTOs e mini-HOWTOs

O verdadeiro espírito da comunidade de Código Aberto cria uma grande coleção de HOWTO/mini-HOWTO. Esses arquivos são exatamente - documentos e guias descrevendo como deve ser feito algo. Caso você decida instalar os HOWTO, eles serão instalados no diretório */usr/doc/Linux-HOWTOs/* e os mini-HOWTOs em */usr/doc/Linux-mini-HOWTOs/*.

A mesma série de pacotes inclui uma coleção de pacotes de FAQs, que é uma abreviação para Frequently Asked Questions (em português seria Perguntas mais Frequentes)

Estes documentos são escritos no estilo de *Perguntas e Respostas*. Os FAQs podem ser muito útil quando se procura algo onde se queira ter *respostas rápidas* para algo. Caso você decida instalar os FAQs durante a instalação, você os encontrará instalados do diretótio */usr/doc/Linux-FAQs/*.

Esses arquivos são muito úteis quando você está com alguma dúvida e não sabe como resolver o problema. Cobrem uma grande variedade de assuntos, mas não cobrem os assuntos de maneira detalhada. Bom material!

2.2 Ajuda Online

Além da documentação que é fornecida com a instalação do Sistema Operacional Slackware Linux, existe uma grande variedade de documentação disponível que você pode obter para aprender.

2.2.1 O Site Oficial e os Fóruns de Ajuda

- Site Oficial do Slackware

  O site Oficial do Slackware Linux algumas vezes fica desatualizado, mas contêm uma grande variedade de informações importantes sobre as últimas versões do Slackware. Existia um fórum ativo de ajuda. Porém manter o fórum começou a dar muito trabalho, então o Pat cancelou o fórum.

  Pode-se encontrar algumas informações antigas desse fórum em www.userlocal.com/phorum. Após o cancelamento do fórum do slackware no site www.slackware.com, diversos outros sites ofereceram-se para possuir o fórum para Slackware, então o Pat endossou o www.linuxquestions.org como sendo o fórum oficial para o Slackware Linux.

2.2.2 Suporte por E-mail

A todos que compram o conjunto oficial de CDs do projeto é oferecido suporte livre via e-mail pelo desenvolvedor. Por favor, isso que está sendo dito, deve ser mantido entre os colaboradores (e a grande maioria dos usuários) do Slackware *A velha escola*. Isso significa que nós preferimos ajudar àqueles que têm um interesse sincero e estão dispostos a ajudar no processo. Nós faremos sempre o melhor para ajudar quem nos envia um e-mail. Entretanto, Por favor verifique a documentação e o site (especialmente os FAQs e talvez algum fórum listado abaixo) antes de enviar um e-mail. Você pode iniciar sua busca nas respostas simples, seria menos e-mail que teríamos que responder, obviamente nós ajudamos de uma forma melhor àqueles que precisam de nossa ajuda.

O endereço de e-mail para suporte técnico é: support@slackware.com. Outros endereços de e-mail e informações de contato são listados no site oficial.

Lista de E-mail do Projeto Slackware Linux Existem algumas listas de e-mail disponíveis em forma de sumário ou formulário normal. Verifique as instruções para saber como se inscrever.

Para inscrever em uma lista de e-mail, envie uma mensagem para:

majordomo@slackware.com

com a frase *subscribe* `[nome da lista]` no corpo da mensagem. A descrição das listas está abaixo (use o nome abaixo para o inscrever-se na lista).

Os arquivos da lista de e-mail podem ser encontrador no site do Slackware em:

http://slackware.com/lists/archive/

slackware-announce

A lista de e-mail slackware-announce é utilizada para anunciar novas versões, principais atualizações e outras informações em geral.

slackware-security

A lista de e-mail slackware-security informa assuntos relacionados à segurança. Exploits e outras vulnerabilidades que pertençam diretamente ao Slackware são postadas na lista imediatamente.

As listas estão disponiveis no formato de sumário. Isto é, ao invés de você receber várias mensagens durante o dia, você receberá apenas uma mensagem com todo o conteúdo criado no dia. A lista de e-mail do Slackware não permite que usuários postem mensagem, e as listas possuem um tráfego baixo, muitos usuários encontram pouca vantagem em receber o sumário da lista. Caso você queira, ainda estão disponíveis para inscrição a lista slackware-announce-digest ou slackware-security-digest.

2.2.3 Sites não-oficiais e Fóruns de ajuda

Sites Google (www.google.com)

O mestre do Kung-Fu para sites de busca. Absolutamente você, encontra as últimas informações sobre o kernel com o título: Não aceite substitutos.

Google:Linux (www.google.com/linux)

Procura especificamente por Linux

Google:BSD (www.google.com/bsd)

Procura especificamente por BSD. Slackware também é um sistema operacional genérico, assim como o Unix funciona, por isso, você também pode encontrar muitas informações sobre Slackware. Muitas vezes uma busca, em BSD, pode trazer informações técnicas relacionadas com o Linux.

Google:Groups (groups.google.com)

Busca por informações nas mensagens de décadas da Usenet.

userlocal

Um tesouro virtual de conhecimento, bom conselheiro, experiências em primeira mão e artigos interessantes. O primeiro lugar que você ouvirá com freqüência sobre o desenvolvimento de Slackware no mundo.

Recursos Web linuxquestions.org

O fórum oficial de usuários de Slackware.

Fórum Slackware

"*Um local para efetuar download e obter ajuda sobre Linux.*"

[alt.os.linux.slackware wombat.san-francisco.ca.us/perl/fom] FAQ

Outras FAQ

Grupos de usuários (NNTP) A USENET tem sido por muito tempo o lugar onde os geeks buscam ajuda e ajudem uns aos outros. Existem poucos grupos dedicados ao Slackware Linux, mas tendem a serem preenchidos com as grandes base de conhecimento de outras pessoas.

alt.os.linux.slackware

alt.os.linux.slackware, mais conhecida como aols, (não confunda com a *AOL!*) é um dos lugares mais ativos para encontrar ajuda sobre problemas com o Slackware. Como cada grupo, alguns participam *(ativamente)* com experiência pelo fato de ajudar bastante. Aprenda a ignorar trolls e identifique ajuda de pessoas que realmente utilizam esse recurso.

Capítulo 3

Instalação

Antes que você possa usar o Slackware Linux, será necessário você obtê-lo e instalá-lo. Obter o Slackware é fácil, você pode compra-lo ou efetuar o download gratuíto através da Internet. A instalação também é muito fácil, desde que você tenha conhecimento básico sobre seu computador e esteja disposto a aprender algumas coisas. O programa de instalação o conduzirá passo-a-passo. Devido ao programa de instalação você instalará rapidamente o sistema. O fato é que o Slackware possui um programa de instalação mais rápido em relação à qualquer outra distribuição Linux.

3.1 Obtendo o Slackware

3.1.1 A caixa com os discos Oficiais

O conjunto de CDs oficial do Slackware Linux CD está disponível na Slackware Linux, Inc. O jogo consiste em 4 discos. O primeiro disco contêm todo o software necessários para um sistema básico e o gerenciados de janelas X. O segundo CD é um disco adicional; isto é, um CD bootável que instala na RAM uma instalação provisória, caso seja necessário efetuar uma manutenção de reparo na máquina. Este CD possui alguns pacotes, tais como os ambientes gráficos KDE e GNOME. Alguns pacotes novos são inclusos no segundo CD incluindo a série de pacotes *extra*, que são pacotes não essenciais ao sistema. O terceiro e o quarto CD inclui todos os códigos fontes de todos os programas do Slackware, junto com a edição original desse livro.

Também é possível comprar um kit contendo os 4 discos e uma cópia deste Livro, dessa forma você poderá se exibir como um verdadeiro geek. A compra dos CDs estão disponíveis a um custo bem acessível.

A maneira preferida para a compra desses produtos é na loja oficial do Slackware.

- http://store.slackware.com

  Você tambêm pode solicitar informações por e-mail.

Informações de Contato na Slackware Linux, Inc.

Method	**Contact Details**
Telephone	1-(925) 674-0783
Site	http://store.slackware.com
Email	orders@slackware.com
Postal	1164 Claremont Drive, Brentwood, CA 94513

3.1.2 Através da Internet

Slackware Linux está totalmente Livre na Internet. Você pode enviar suas perguntas por email, mas a prioridade será às perguntas referentes à compras do kit oficial. Como dito anteriormente, nós recebemos muitos e-mails, e o tempo é muito limitado para poder atender a todos. Antes de enviar um e-mail para suporte primeiramente leia a ajuda.

O site oficial do Projeto Slackware Linux Project é o endereço:
http://www.slackware.com/
A localização do FTP primário para o Slackware Linux é:
ftp://ftp.slackware.com/pub/slackware/
O ftp disponível no site, está aberto para uso geral, não existe limite de banda. Mas por favor, considere a opção de utilizar um mirror mais próximo à você para efetuar o download do Slackware. A lista completa dos mirros podem ser encontradas no site http://www.slackware.com/getslack.

3.2 Requisitos para Instalação

Uma instalação mínima do Slackware, necessita da seguinte configuração:

Hardware	**Necessário**
Processador	586
Memória	32 MB RAM
Espaço em Disco	1GB
Drive de CD	4x CD-ROM

Caso você tenha um CD bootável, você não irá precisar de um disquete. É claro que se você não possuir uma unidade de CD-ROM, você precisará de um disquete e uma unidade de disquete para fazer a instalação via NFS através de uma rede. Além disso também é necessário uma placa de rede para esse tipo de instalação. Para maiores informações consulte a seção NFS.

O espaço em disco necessário, é um pouco complicado especificar. O espaço recomendado para uma instalação mínima é de 1GB, mas caso você faça uma instalação completa você irá precisar cerca de 2Gb de espaço em disco e mais um espaço para armazenar seus arquivos pessoais. A maioria dos usuários não fazem a instalação completa. O fato é que muitos usuários instalam o Slackware em um disco realmente pequeno com 100Mb de espaço.

O Slackware pode ser instalado em sistemas com pouca memória RAM, discos rígidos pequenos e CPUs com pouco poder de processamento, porém isso necesita de um pouco de trabalho. Se este for o seu caso, leia o arquivo *LOWMEM.TXT* localizado na árvore da distibuição, para obter algumas dicas úteis.

3.2.1 As Séries do Software

Por razões de simplicidade, os softwares (pacotes) que acompanham o Slackware são divididos em séries. Essas séries são chamadas de *conjunto de discos* devido terem sido planejadas para a instalação do sistema através de disquetes, porém hoje essas séries são usadas basicamente para categorizar os pacotes incluso no Slackware. Hoje não é mais possível fazer a instalação apenas com disquetes.

A seguir você encontra uma breve descrição de cada série de pacotes.

Série dos Pacotes

Série	Conteúdo
A	Base do Sistema. Contêm uma grande parte dos softwares utilizados para se ter um sistema
AP	Diversos aplicativos que não necessitam de um Servidor X.
D	Ferramentas de desenvolvimento de programas. Compiladores, debugadores, interpretadores
E	GNU Emacs.
F	FAQs, HOWTOs e outros tipos de documentação.
GNOME	O ambiente gráfico GNOME.
K	O código fonte do kernel do Linux.
KDE	O ambiente gráfico K. Um ambiente gráfico que possui muitas características visuais em com
KDEI	Pacotes de internacionalização para o ambiente KDE.
L	Bibliotecas. Bibliotecas dinâmicas lincadas necessárias para muitos outros programas.
N	Programas de Rede. Daemons, programas de e-mail, telnet, leitor de news e alguns outros.
T	Sistema de formatação de documentos teTeX.
TCL	Ferramenta de Linguagem de comando. Tk, TclX, e TkDesk.
X	Sistema base para o X Window.
XAP	Aplicações gráficas que não fazem parte de um ambiente de desktop principal (como por exe
Y	Jogos de console BSD

3.2.2 Métodos de Instalação

Disquete

Quando era possível, O Slackware Linux era instalado com apenas dois disquetes, porém o tamanho dos pacotes de alguns softwares cresceram (especificamente alguns softwares) então houve o abandono forçado da instalação via disquete. Até a versão 7.1, era possível fazer uma parte da instalação utilizando disquetes. As séries A e N podiam ser totalmente instaladas, o que fornecia um sistema básico para instalar o restante da distribuição. Caso você for utilizar um disquete para instalar (em um equipamento mais velho), é recomendável que você utilize uma versão mais antiga. Para esse tipo de equipamento é recomendado as versões 4.0 (a mais popular para isso) e a 7.0 do Slackware.

Observe que o disquete ainda é utilizado para realizar o boot, quando isso não é possível através de uma unidade de CD-ROM, ou quando se utiliza a instalação via NFS.

CD-ROM

Se você tiver um CD bootável, criado pela Slackware Linux, Inc. (veja a seção Obtendo o Slackware) a instalação ficará mais fácil para você. Caso contrário, você poderá utilizar um disquete de boot. Caso, você tenha um hardware específico, um disco de boot específico poderá ser utilizado.

A partir da versão 8.1, um novo método de criação de CDs de BOOT foi adotado, que não trabalha com algumas BIOS (esse problema é enfrentado por todas distribuições Linux atualmente). Se este for o seu caso, recomendamos que você utilize um disquete.

Veja a seção 3.2.3 e a seção 3.2.5 fornecem informações de como criar um disquete de boot, caso seja necessário.

NFS

NFS (Sistemas de Arquivos de Rede) é uma maneira de utilizar sistemas de arquivos em máquinas remotas. A instalação via NFS permite que você instale o Slackware à partir de uma outra máquina de sua rede. A máquina que você estiver utilizando para a instalação precisa ser configurada de modo a exportar a árvore de distribuição do Slackwware para a máquina que você está instalando. Isto, naturalmente, requer um certo conhecimento do NFS, que é coberto na seção Network File System (NFS).

Também é possível executar o NFS através de um método como o PLIP (sobre uma porta paralela), SLIP, e PPP (exceto sobre uma conexão via modem). Entretanto, é recomendável que você utilize uma placa de rede. Apesar de tudo, instalar um sistema através de uma porta paralela é algo muito lento.

3.2.3 Disco de Boot

O disco de boot é um disquete utilizado para iniciar a instalação. Ele contêm uma imagem compactada do Kernel que é utilizada para controlar o hardware durante a instalação. Ao menos que você esteja utilizando um CD, este disco é necessário. Os discos de boot são localizados no diretório *bootdisks/* da árvore da distribuição.

Existem diversos discos de boot que você pode utilizar (são aproximadamente 16). A lista completa dos discos de boot e a descrição completa de cada um são encontradas no arquivo *bootdisks/README.TXT* localizado na árvore da distribuição. Entretanto, a maioria das pessoas utilizam a imagem *bare.i* (para dispositivos IDE ou *scsi.s* (para dispositivos SCSI) para o boot.

Veja a seção 3.2.3 - Disco de Boot para instruções de como fazer um disco de boot.

Após o boot, um prompt de comando solicitará que você insira o disco de root.

3.2.4 Disco de Root

Os discos de root contêm o programa de instalação e um sistema de arquivos que é utilizado durante a instalação. Também são necessários. As imagens do disco de root são encontradas na árvore da distribuição. Você terá que fazer dois discos de root, *install.1* e *install.2*. Aqui, você também pode encontrar os discos *network.dsk*, *pcmcia.dsk*, *rescue.dsk*, e *sbootmgr.dsk*.

3.2.5 Disco Complementar

Um disco complementar é necessário caso você utilize uma instalação via NFS ou se você tiver um cartão PCMCIA. Os discos complementares são encontrados no diretótio raíz da árvore de distribuição, com os nomes *network.dsk* e *pcmcia.dsk*. Recentemente, outros discos complementares como *rescue.dsk* e *sbootmgr.dsk* foram adicionados. O disco de restauração é uma imagem pequena que funciona com 4MB de RAM. Inclui alguns utilitários básicos de rede e o editor vi para reparos rápidos. O disco *sbootmgr.dsk* é utilizado para carregar outros dispositivos. Use este disco se o seu CD-ROM bootável não quiser dar boot com os CDs do Slackware. Ele pedirá outras opções de boot e pode ser uma forma conveniente de contornar uma BIOS com bugs.

O disco de root irá instruí-lo a usar discos complementares quando ele for carregado.

3.2.6 Criando os discos

Caso tenha escolhido uma imagem de boot, você necessita de um disquete. Dependendo do sistema que está usando para criar os discos, o processo será diferente. Se você estiver utilizando um Linux (ou um outro sistema Unix) você utilizará o comando *dd*(1). Suponhamos que você esteja usando a imagem *bare.i* e a sua unidade de disquete é o */dev/fd0*, o comando que você deverá utilizar para criar um disquete com a imagem *bare.i* é:

```
$ dd if=bare.i of=/dev/fd0
```

Se estiver utilizando um sistema operacional Microsoft, você precisará do programa *RAWRITE.EXE*, que está incluso no mesmo diretório das imagens de disquete. Suponhamos que esteja usando a imagem *bare.i* e o seu dispositivo de disquete é a unidade *A:*, abra o prompt do DOS e digite:

```
C: rawrite a: bare.i
```

3.3 Particionamento

Após a inicialização com a sua mídia preferida, será necessário particionar o seu disco rígido. A partição do disco é onde o sistema de arquivos do Linux será criado e onde o Slackware será instalado. Recomendamos a criação de, no mínimo, duas partições; uma para o seu sistema de arquivos root (/) e uma para a área de troca (swap).

Quando terminar o carregamento do disco de root, aparecerá um prompt de login. Entre no sistema como root (sem senha). No prompt do shell, execute *cfdisk*(8) ou *fdisk*(8). O programa *cfdisk* possui uma interface mais amigável do que o programa *fdisk*, mas lhe faltam alguns recursos. Segue abaixo uma rápida explicação do programa *fdisk*.

Comece executando o *fdisk* para o seu disco rígido. No Linux, em vez dos discos rígidos possuírem letras de drive, eles são representados por um arquivo. O primeiro disco rígido IDE (primário master) é */dev/hda*, o primário slave é */dev/hdb*, e assim por diante. Discos SCSI seguem o mesmo sistema, mas no formato */dev/sdX*. Inicie o *fdisk* informando o seu disco rígido:

```
# fdisk /dev/hda
```

Como em todos os bons programas para o Unix, o *fdisk* mostra um prompt (pensou que haveria um menu, não foi?). A primeira coisa que você deve fazer é examinar o seu esquema de partições atual. Faça isso digitando p** no prompt do *fdisk*:

```
Command (m for help): p
```

Serão exibidos todos os tipos de informações sobre o seu sistema de partições atual. A maioria das pessoas seleciona uma unidade livre para a instalação e então remove as partições existentes para dar espaço para as partições do Linux.

AVISO: É EXTREMAMENTE IMPORTANTE FAZER UMA CÓPIA DE SEGURANÇA (BACKUP) DE TODAS AS INFORMAÇÕES QUE VOCÊ QUEIRA MANTER, ANTES DE REMOVER A PARTIÇÃO ONDE ELAS ESTÃO ARMAZENADAS.

Não existe uma forma fácil de recuperar dados de uma partição removida, portanto sempre faça uma cópia de segurança (backup) antes de alterá-las.

Na tabela de informações da partição há um número da partição, o tamanho, e o tipo. Há mais informações, mas não se preocupe com isso por enquanto. Removeremos todas as partições deste drive para criar as do Linux. Digitamos **d** para removê-las:

Command (m for help): d
Partition number (1-4): 1

Repita este processo para cada uma das partições. Após removê-las, estaremos prontos para criar as do Linux. Decidimos criar uma partição para o nosso sistema de arquivos root e uma para o swap. Observe que os esquemas de particionamento do Unix são a causa de acaloradas discussões, e que a maioria dos usuários lhe dirá qual a melhor forma. No mínimo, crie uma partição para o / e uma para swap. Com o tempo, você desenvolverá uma metodologia que funcione bem para você.

Eu uso dois esquemas básicos de particionamento. O primeiro é para um desktop. Faço 4 partições, /, */home*, */usr/local*, e swap. Assim posso re-instalar ou atualizar tudo que está instalado em / sem remover os meus arquivos de dados no /home ou os meus aplicativos personalizados compilados no */usr/local*. Em servidores, geralmente substituo a partição */usr/local* por uma partição */var*. Muitos servidores diferentes armazenam informações nesta partição e mantê-la separada do / traz benefícios quanto ao desempenho. Por enquanto, vamos ficar com apenas estas duas partições: / e swap.

Agora podemos criar as partições com o comando **n**:

```
Command (m for help):n
Command action
e   extended
p   primary partition (1-4)
p
Partition number (1-4)}: 1
First cylinder (0-1060, default 0): 0

Last cylinder or +size or +sizeM or +sizeK (0-1060, default 1060):+64M
```

Assegure-se de ter criado partições primárias. A primeira partição será a nossa partição swap. Diremos ao fdisk para tornar a partição número 1 uma partição primária. Ela começará no cilindro 0 e, para o cilindro final, digitamos +64M. Isto criará uma partição de 64 megabytes para swap. (Na verdade, o tamanho da partição de swap depende da quantidade de RAM do seu sistema. Por convenção se usa uma partição de swap com o dobro do tamanho da RAM do seu sistema.) Então, definimos a partição primária número 2 começando no primeiro cilindro disponível e indo até o final do drive.

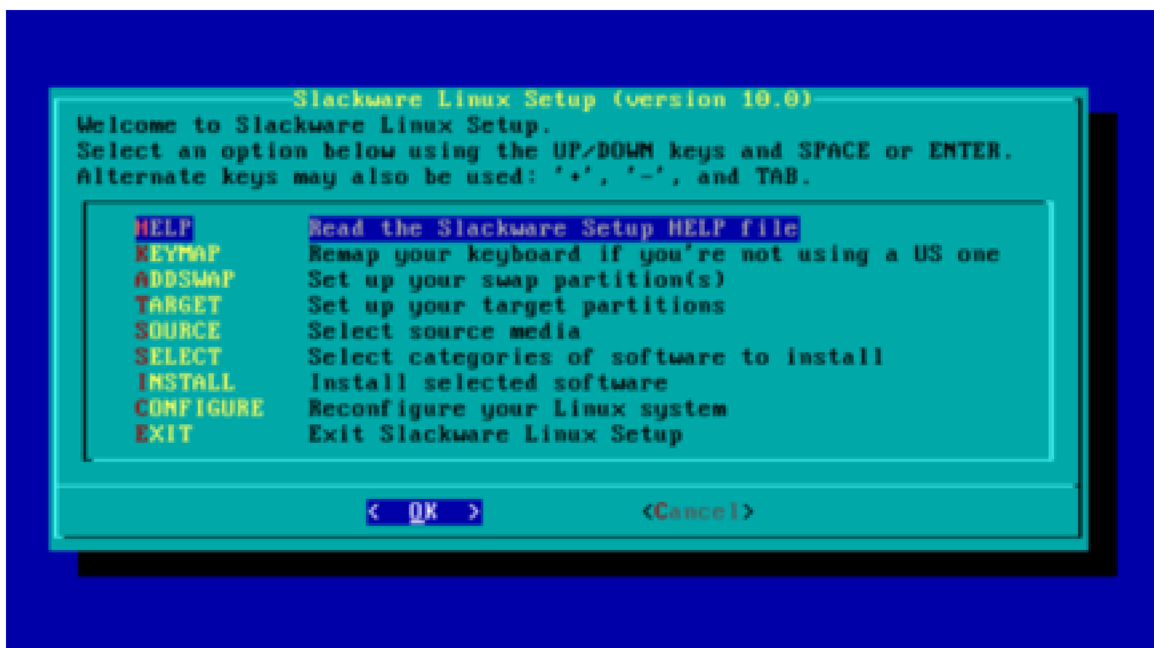
```
Command (m for help): n
Command action
e   extended
p   primary partition (1-4)
```

```
p
Partition number (1-4): 2
First cylinder (124-1060, default 124):124
Last cylinder or +size or +sizeM or +sizeK (124-1060, default 1060):}1060
```

Está quase pronto. Precisamos alterar o tipo da primeira partição para tipo 82 (Linux swap). Digite **t** para alterar o tipo, selecione a primeira partição, e digite *82*. Antes de gravar as suas alterações em disco, dê uma última olhada na nova tabela de partições. Use o **p** no *fdisk* para exibir a tabela de partições. Se tudo parecer estar bem, digite **w** para salvar as suas alterações e sair do *fdisk*.

3.4 O programa setup

Uma vez criada as partições, você está pronto para instalar o Slackware. O próximo passo da instalação é executar o programa *setup*. Para executá-lo simplesmente digite *setup* no prompt. O *setup* possui um sistema de menu na qual permite que você instale os pacotes do Slackware e configure seu sistema.



O processo de instalação é algo do tipo: Selecione as opções de acordo como elas são listadas no programa *setup*. (É claro que você pode mudar a ordem de escolha das opções, não seguindo passo-a-passo o programa.) Os itens do menu são selecionados usando as teclas seta para cima e seta para baixo, e os botões *Okay* e *Cancel* são usados para proseguir ou para voltar. Como alternativa, você pode escolher a opção desejada, teclando a letra correspondente destacada do menu para cada opção. Para selecionar as opções (estas são indicadas com *[X]*) utiliza-se a barra de espaço.

É claro que você pode ver mais descrições na seção *help* do *setup*, porém acreditamos que o seu tempo vale dinheiro.

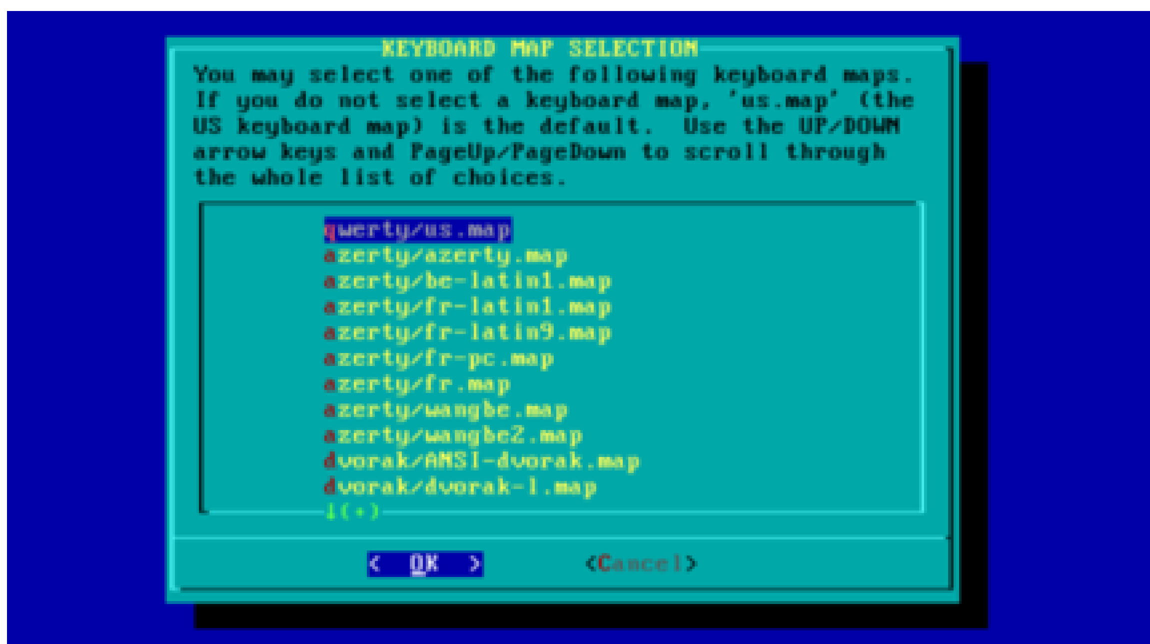
3.4.1 HELP

Caso seja a primeira vez que você esteja instalando o Slackware, você pode dar uma olhada na tela de ajuda. Assim terá uma descrição de cada parte do programa *setup* (assim como estamos descrevendo) e instruções para navegação e para o restante da instalação.

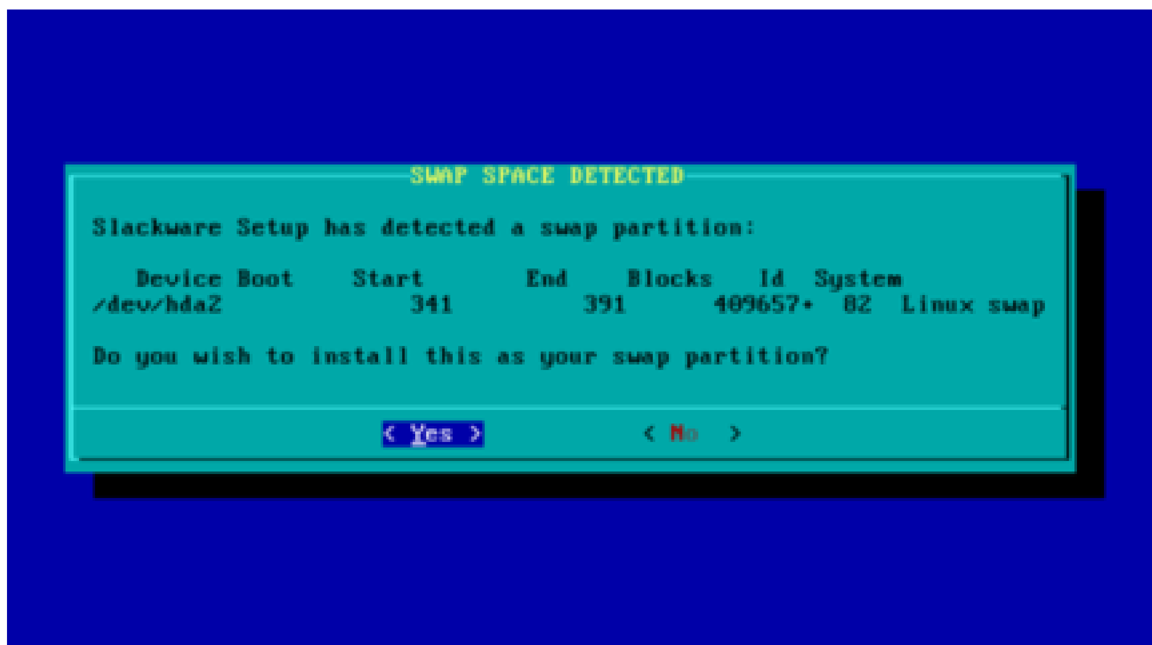


3.4.2 KEYMAP

Caso necessite de um teclado diferente do padrão americano *qwerty*, você deverá alterá-lo nessa seção. Existe uma grande quantidade de modelos de teclado que poderá escolher.

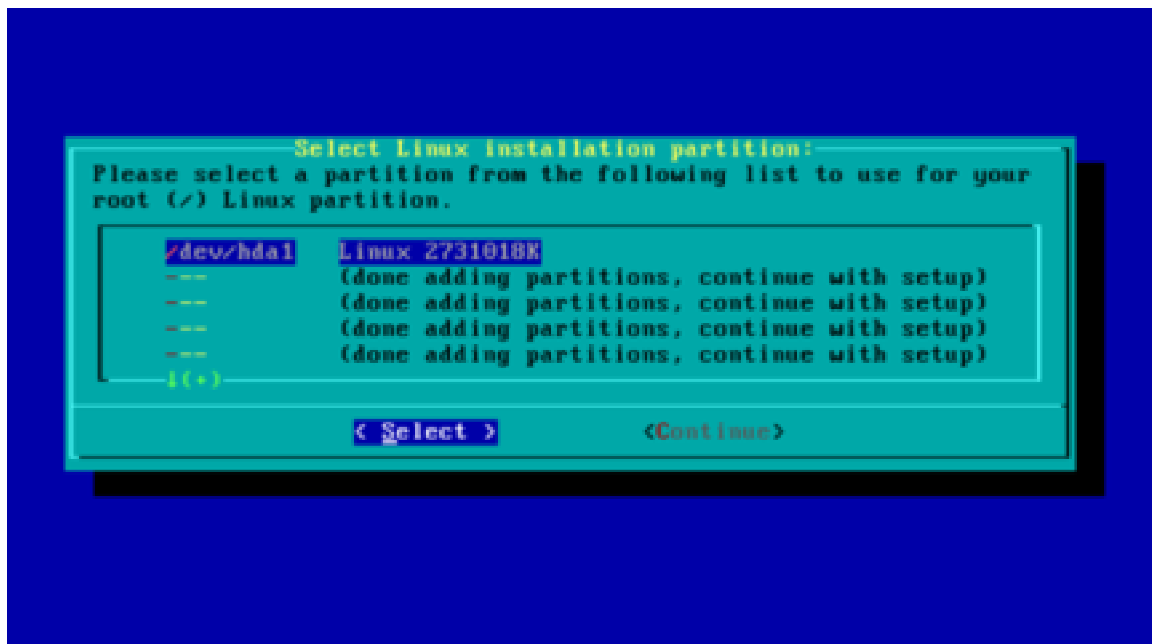


3.4.3 ADDSWAP



Caso você tenha criado uma partição de swap (voltar à seção 3.3), é aqui que você deverá habilitá-la para utilizar. As partições de swap existentes em seu disco serão exibidas automaticamente, permitindo que você selecione, pelo menos uma, para formatar e habilitar.

3.4.4 TARGET



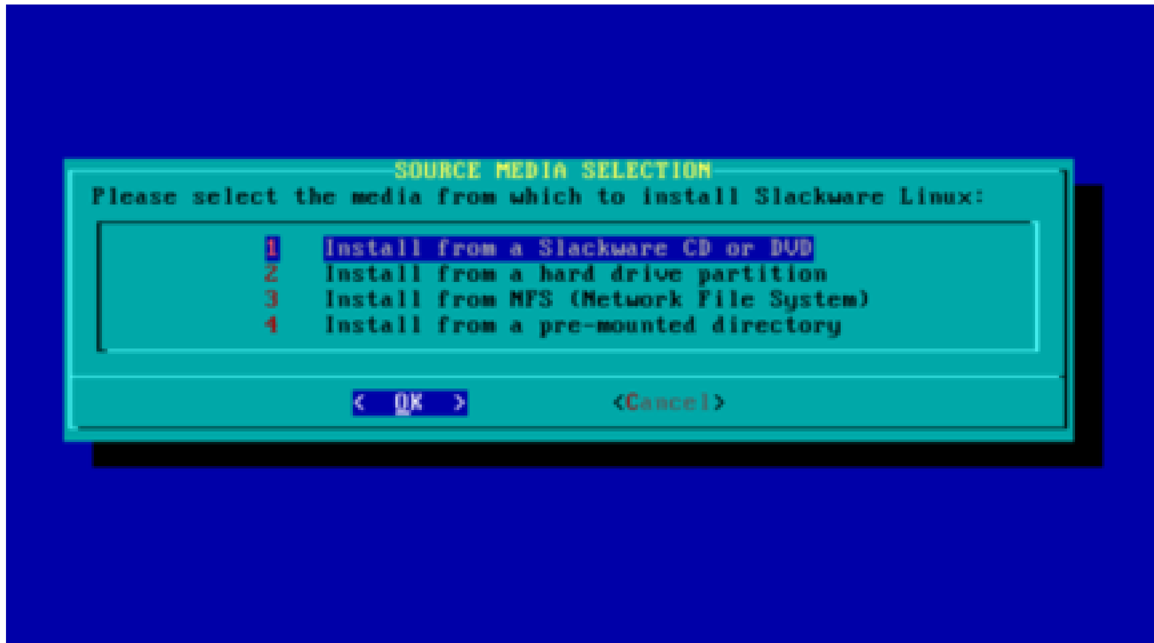
Na seção target você formatará suas partições e mapeará os pontos de montagens dos sistemas de arquivos. Uma lista com as partições de seu disco rígido são mostradas. Para cada partição, poderá escolher formatar ou não. Dependendo do kernel utilizado, você poderá escolher os seguintes sistemas de arquivos reiserfs (padrão), ext3, ext2, jfs e xfs. A maioria das pessoas utilizam reiserfs ou ext3. Em um futuro próximo nós poderemos ter o suporte à reiserfs4.

A partição a ser selecionada é onde o sistera root (/)deverá ser instalado. Após isso, você

poderá escolher e mapear cada partição para cada sistema de arquivos de acrodo com a sua escolha. (Por exemplo, você pode querer que a sua terceira partição */dev/hda3*, seja o sistemas de arquivos para o home. Isto é apenas um exemplo; crie os pontos de montagens de acordo com o seu gosto.)

3.4.5 SOURCE

A seção Source é onde selecionamos a mídia a partir da qual o Slackware será instalado. Atualmente, há quatro opções disponíveis. São elas: CD-ROM, uma partição no disco rígido, NFS, ou um diretório pré-montado.



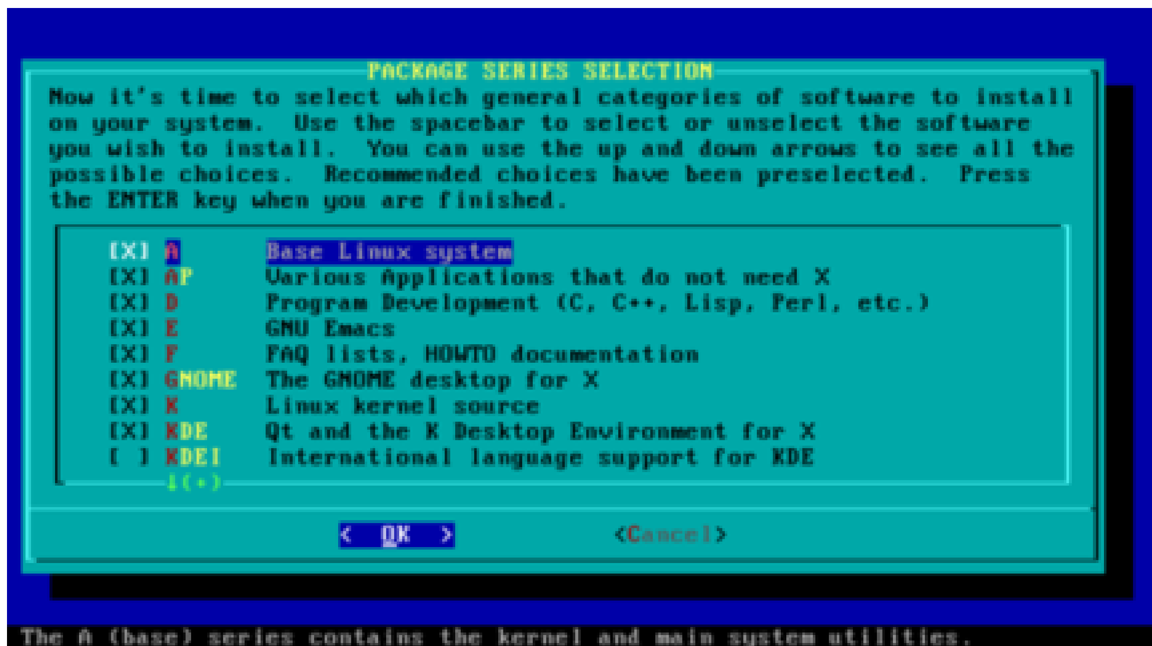
A opção CD-ROM permite a instalação a partir de um CD-ROM. Há a opção de procurar automaticamente um drive de CD-ROM ou exibir uma lista da qual pode-se escolher o tipo do drive. Assegure-se de ter o CD do Slackware no seu drive antes de começar a busca.

A opção NFS pede informações da sua rede e de seu servidor NFS. O servidor NFS já deve estar configurado. Note também que não é possível usar nomes de máquina (hostnames), deve-se usar endereços IP tanto para a sua máquina, quanto para o servidor NFS (não há um serviço de resolução de nomes no disco de instalação). Naturalmente, deve ser usado o disquete *network.dsk* para dar suporte para a sua placa de rede.

O diretório pré-montado oferece maior flexibilidade. Use este método para instalar a partir de dispositivos como Jaz disks, pontos de montagem NFS sobre PLIP, e sistemas de arquivos FAT. Monte o sistema de arquivos em um local de sua escolha antes de executar a instalação, e então especifique esse local aqui.

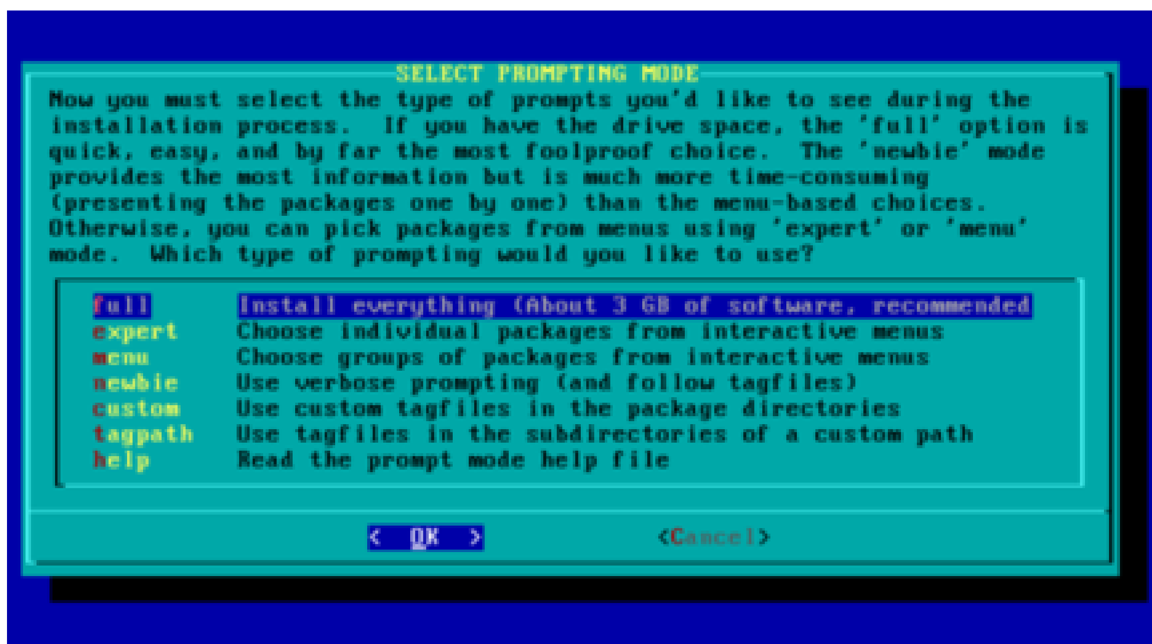
3.4.6 SELECT

A seção Select permite escolher as séries de software que desejamos instalar. Essas séries são descritas na seção 3.2.1. Note que é obrigatória a instalação da série A para ter um sistema base funcional. Todas as outras séries são opcionais.



3.4.7 INSTALL

Considerando que tenhamos passado pelas seções *"target"*, *"source"*, and *"select"*, a seção *install* permite selecionar pacotes dentre as séries de software escolhidas. Se não, pedirá que você volte e complete as outras seções do programa de instalação. Esta seção permite escolher dentre seis diferentes métodos de instalação: *//full*, *newbie*, *menu*, *expert*, *custom*, e *tag path*.



A opção *full (completo)* instala todos os pacotes das séries escolhidas na seção *select*. Não há mais perguntas. Este é o método de instalação mais fácil, já que não é necessário tomar quaisquer decisões quanto aos pacotes a instalar de fato. É claro que esta opção também é a que ocupa maior espaço no disco rígido.

A próxima opção é *newbie (novato)*. Esta opção instala todos os pacotes requeridos pelas séries selecionadas. Em todos os outros pacotes há um prompt onde podemos selecionar *Yes*

(sim), *No (não)*, ou *Skip (pular)*. Yes e No fazem o óbvio, enquanto Skip pula para a próxima série de software. Além disso, é exibida uma descrição e o tamanho de cada pacote para ajudar a decidir se o mesmo é necessário. Recomendamos esta opção para os novos usuários, já que ela assegura que os pacotes desejados sejam instalados. Contudo, ela é um pouco mais lenta devido às perguntas.

Menu é uma versão mais rápida e mais avançada da opção newbie. Em cada série, aparece um menu, a partir do qual pode-se selecionar todos os pacotes não requeridos que se deseja instalar. Os pacotes requeridos não aparecem nesse menu.

Para os usuários mais avançados, a instalação oferece a opção *expert*. Isto permite total controle sobre quais pacotes são instalados. É possível remover da seleção pacotes que são absolutamente necessários, o que resulta em um sistema entrar em colapso. Por outro lado, é possível controlar exatamente o que entra em seu sistema. Simplesmente selecione os pacotes de cada série que deseja que sejam instalados. Isto não é recomendado para os novos usuários, já que é muito fácil dar um tiro no próprio pé.

As opções *custom* e *tag path* também são para usuários avançados. Estas opções permitem a instalação com base em tag files personalizados criados na árvore da distribuição. Isto é útil para se instalar um grande número de máquinas rapidamente. Para maiores informações sobre o uso de tag files, veja a seção 18.4.

Após escolher o seu método de instalação, uma coisa ou outra abaixo vai acontecer. Se tiver selecionado full ou menu, uma tela com um menu aparecerá, permitindo a seleção dos pacotes a ser instalados. Se tiver selecionado full, os pacotes começarão a ser instalados no destino imediatamente. Se tiver selecionado newbie, os pacotes serão instalados até que se alcance um pacote opcional. Observe que é possível que o espaço em disco acabe durante a instalação. Se selecionar pacotes demais para a quantidade de espaço livre no dispositivo alvo, você terá problemas. É mais seguro selecionar alguns programas e adicionar mais depois, se necessário. Isto pode ser feito facilmente usando as ferramentas de gerenciamento de pacotes do Slackware. Para maiores informações, veja a seção 18.

3.4.8 CONFIGURE

A seção configure permite fazer algumas configurações básicas do sistema, agora que os pacotes já foram instalados. O que você vê depende em grande parte de que programas foram instalados. Contudo, sempre será exibido o conteúdo abaixo:

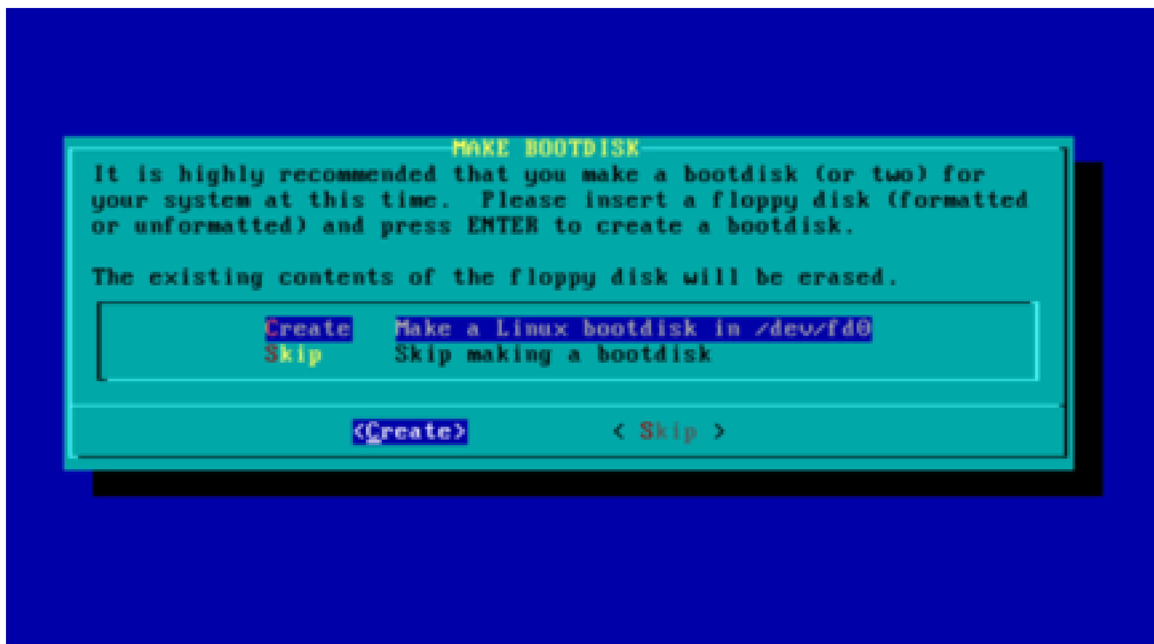
Kernel selection (Seleção do kernel)

Aqui você deverá escolher um kernel para ser instalado. Você pode instalar o kernel do disco de boot usado na instalação, do CD-ROM do Slackware, ou de outro disquete que você (sempre se antecipando) tiver preparado. Ou você pode decidir pular, sendo que então o kernel padrão será instalado e o processo continuará.



Make a boot disk (Fazer um disco de boot)

Provavelmente será uma boa idéia fazer um disco de boot para usar no futuro. Você terá a opção de formatar um disquete e, então, criar um dos dois tipos de disco de boot. O primeiro tipo, *simple (simples)*, simplesmente (como é de se esperar) grava um kernel no disquete. Uma opção mais flexível (e altamente recomendada) é *lilo*, que, é claro, cria um disco de boot do lilo. Veja LILO na seção 7.1 para maiores informações. É claro, você também pode decidir simplesmente *continuar*, de modo que não será criado um disco de boot.



Modem

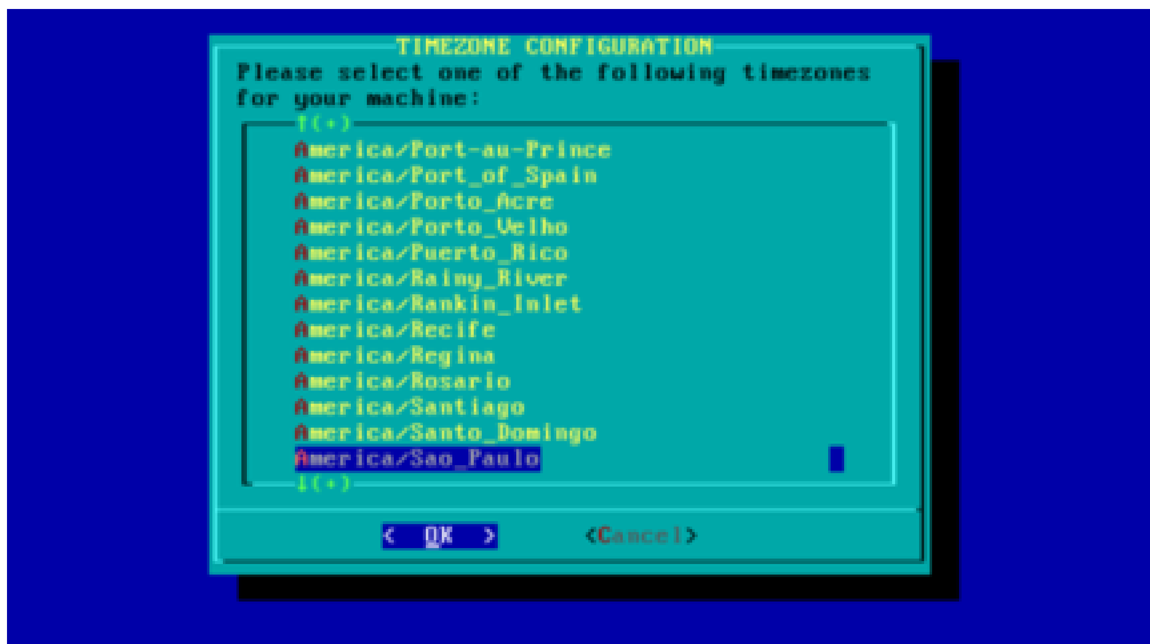
Serão solicitadas informações sobre o seu modem. Mais especificamente, se você possui um modem, e, se possuir, em qual porta serial ele está.



As próximas sub-seções de configuração podem aparecer ou não, dependendo se os seus pacotes correspondentes tiverem sido instalados.

Timezone (Fuso horário)

Aqui é bem direto: será perguntado em qual fuso horário você está. Se estiver no horário Zulu, sentimos muito; a lista (extremamente longa) está em ordem alfabética, e você está no final dela.



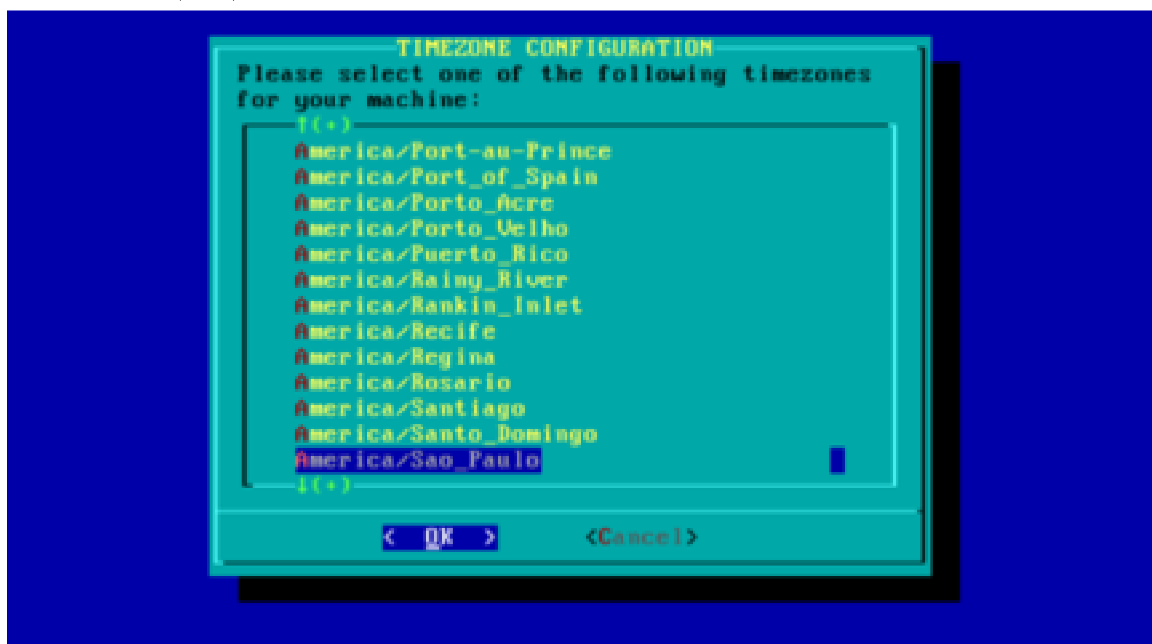
Mouse

Esta sub-seção simplesmente pergunta que tipo de mouse você tem, e se você quer ter suporte ao mouse no console com o *gpm*(8) habilitado na inicialização.



Hardware clock (Relógio de hardware)

Esta sub-seção pergunta se o relógio de hardware do seu computador está ajustado no Tempo Universal Coordenado (UTC ou GMT). A maioria dos PCs não estão, então provavelmente você deve dizer ”no”(não).



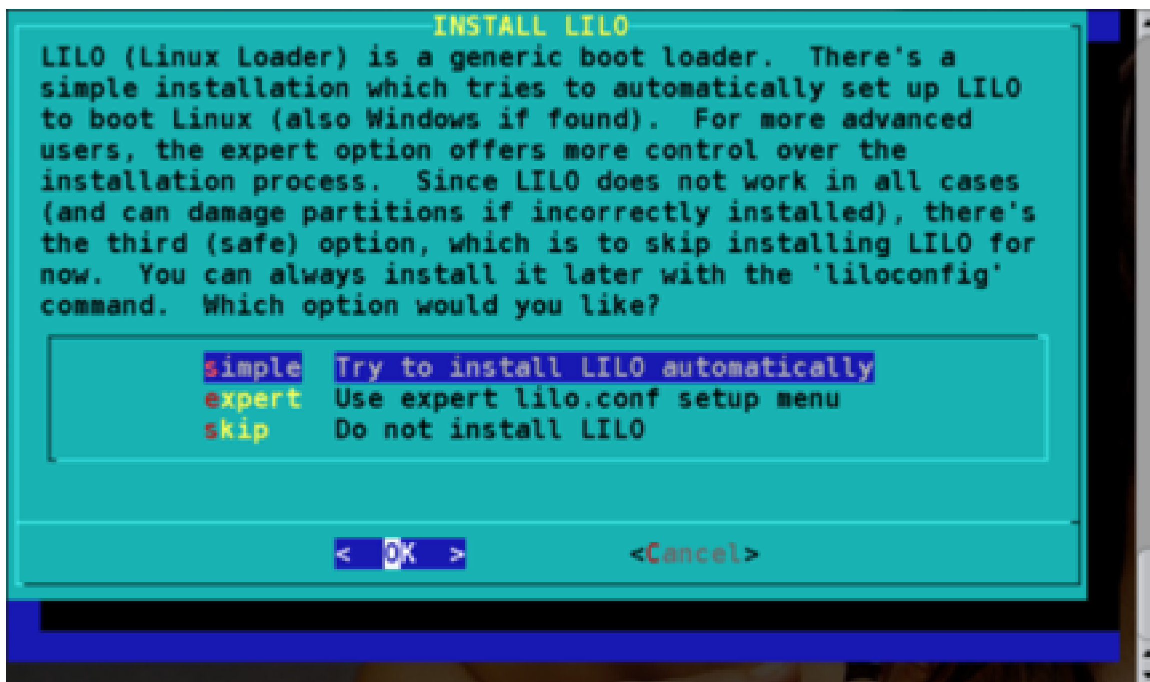
Font

A sub-seção font permite que se escolha dentre de uma lista de fontes personalizadas para o console.



LILO

Aqui é perguntado sobre a instalação do LILO (o LInux LOader; veja a seção 7.1 para maiores informações).



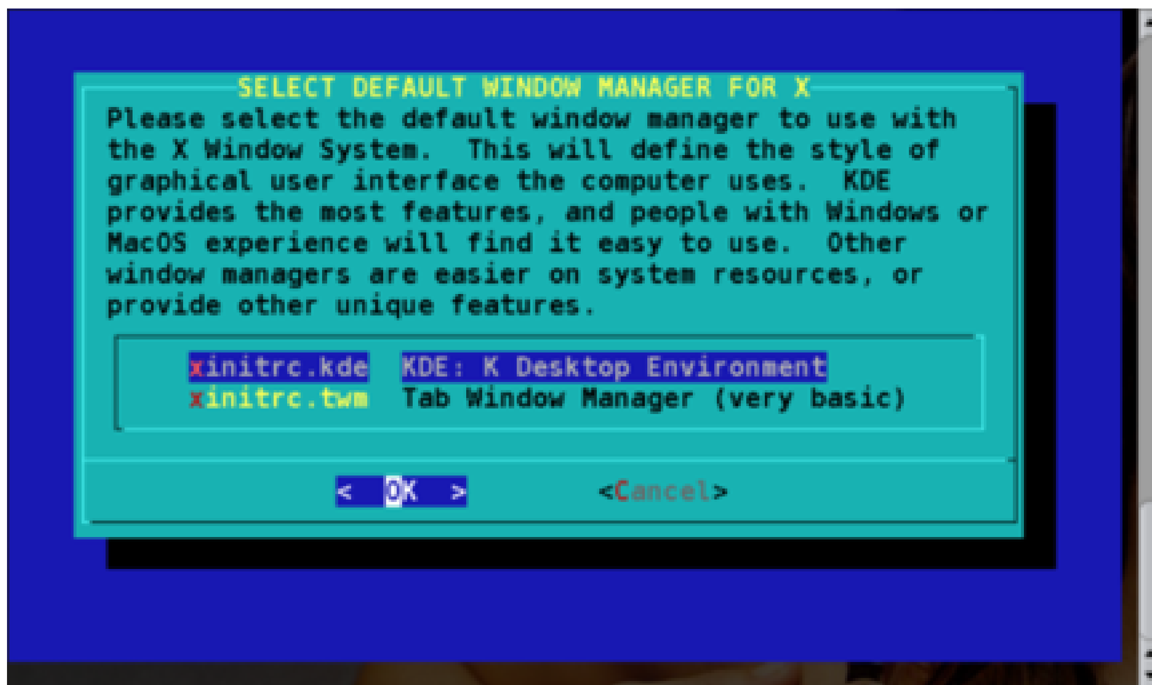
Se o Slackware for o único sistema operacional em seu computador, a opção *simple (simples)* deve funcionar perfeitamente. Se houver dual-boot, a opção *expert* é uma melhor escolha. Veja a seção 7.5 para maiores informações sobre dual-boot. A terceira opção, *do not install (não instalar)*, não é recomendada, a não ser que você saiba o que está fazendo e que tenha uma ótima razão para não instalar o LILO. Se o modo de instalação for "expert", você poderá escolher onde o LILO será instalado. O LILO pode ficar na MBR (Master Boot Record) do seu disco rígido, no superbloco da sua partição raíz do Linux, ou em um disquete.

Network

Na verdade, a sub-seção de configuração network é o *netconfig*. Veja a seção 5.1 para maiores informações.

X Window Manager (Gerenciador de Janelas do X)

Esta sub-seção permite a escolha de um gerenciador de janelas padrão para o X. Veja seção 5.1 para maiores detalhes sobre o X e gerenciadores de janelas.



Não importa quais pacotes tenham sido instalados, a última coisa que será feita na seção configure será perguntar se você quer continuar e criar uma senha para o *root*. Por razões de segurança, provavelmente está será uma boa idéia; todavia, como tudo mais no Slackware, você decide.

Capítulo 4

Configuração do Sistema

Antes de você poder configurar as partes mais avançadas do seu sistema, é uma boa idéia entender como o sistema é organizado e que comandos podem ser utilizados para procurar por arquivos e programas. É bom saber também, se você precisará compilar um kernel otimizado e quais os passos que deverão ser seguidos. Este capítulo irá familiarizá-lo com a organização do sistema e a configuração de arquivos. Assim, você poderá avançar na configuração das partes mais avançadas do sistema.

4.1 Visão Geral do Sistema

É importante entender como um sistema Linux é montado antes de mergulhar nos vários aspectos de sua configuração. Um sistema Linux é significativamente diferente de um sistema DOS, Windows ou de um sistema Macintosh (com a exceção do Mac OS X que é baseado no Unix), mas estas seções o ajudarão a ficar familiarizado com o seu layout, de modo que você facilmente poderá configurar seu sistema para as suas necessidades.

4.1.1 Layout do Sistema de Arquivos

A primeira diferença marcante entre o Slackware Linux e um sistema DOS ou Windows é o sistema de arquivos. Logo de início, não são utilizados letras para os drivers, para identificar as diferentes partições. Sob o Linux, existe apenas um único diretório principal. Você pode relacionar isto com o drive *C:* do ambiente DOS. Cada partição em seu sistema é montado como um diretório dentro do diretório principal. É como se você tivesse um disco rígido expansível.

Chamamos o diretório principal de diretório root (ou raíz), e o mesmo é representado por uma barra (/). Este conceito pode soar estranho, mas isso torna sua vida mais fácil quando você quer adicionar mais espaço. Por exemplo, vamos dizer que você ficou sem espaço em disco no diretório */home*. Muitas pessoas ao instalarem o Slackware criam uma única partição raiz. Bem, como uma partição pode ser montada para qualquer diretório, você só precisa ir até a loja mais próxima e comprar um disco rígido novo e montá-lo como */home*. Você agora colocou mais espaço em seu sistema. E tudo sem precisar mover nada de lugar.

Abaixo, você irá encontrar as descrições dos diretórios mais importantes sob o Slackware.

bin - Programas essenciais para os usuários são armazenados aqui. Isso representa um conjunto mínimo de programas requeridos pelo usuário para a utilização do sistema. Coisas como o shell e os comandos do sistema de arquivos (*ls*, *cp*, e assim por diante) são armazenados aqui. O diretório */bin* também não recebe modificação após a instalação. Se isso acontecer, é na forma de atualização de pacotes que fornecemos.

boot - Arquivos que são utilizados pelo Linux Loader (LILO). Esse diretório também recebe pequenas modificações após a instalação. O kernel é armazenado aqui desde a versão 8.1 do Slackware. Em versões anteriores do Slackware, o kernel era simplesmente armazenado sob o

diretório / , porém a prática comum é montar o kernel e os arquivos relacionados aqui, para facilitar o dual-booting (ou boot duplo).

dev - Tudo no Linux é tratado como arquivo, inclusive dispositivos de hardware como as portas seriais, discos rígidos, e scanners. Em ordem de acessar esses dispositivos, um arquivo especial chamado de dispositivo de conexão tem de estar presente. Todos os dispositivos de conexão são armazenados no diretório */dev*. Você vai descobrir que isso é verdade através dos muitos sistemas operacionais baseados em Unix.

etc - Esse diretório mantém os arquivos de configuração do sistema. Todos os arquivos de configuração do X Window, o banco de dados do usuário, até o sistema de inicialização de scripts. Com o tempo, o administrador do sistema ficará bastante familizado com este diretório.

home - O Linux é um sistema operacional multi-usuário. A cada usuário no sistema é dado uma conta e um diretório único para seus arquivos pessoais. Este diretório é chamado de diretório raíz do usuário. O diretório */home* é fornecido como localização padrão para os diretórios raíz do usuário.

lib - As bibliotecas do sistema que são necessárias para operações básicas do sistema são armazenadas aqui. A biblioteca C, o carregador dinâmico, as bibliotecas ncurses e os módulos do kernel estão entre as coisas aqui armazenadas.

mnt - Este diretório contém pontos de montagem temporários para discos rígidos em operação ou drives removíveis. Aqui você encontrará os pontos de montagem para o seus drives de CD-ROM e de disquete.

opt - Pacotes de software opcionais. A idéia por trás do diretório */opt* é que cada pacote de software instalado no diretório */opt/pacote de software*, que seja fácil de remover depois. O Slackware distribui algumas coisas no diretório */opt* (como o KDE em */opt/kde*), porém você é livre para adicionar qualquer coisa que queira ao diretório */opt*.

proc - Este é um diretório único. Ele não é realmente parte do sistema de arquivos, mas sim um sistema de arquivos virtual que provê acesso as informações do kernel. Vários pedaços de informação que o kernel quer que você conheça são transportados a você através de arquivos no diretório */proc*. Você pode enviar informação para o kernel através desses mesmos arquivos. Tente o comando *cat /proc/cpuinfo*.

root - O administrador do sistema é conhecido como *root* no sistema. O diretório raíz do usuário *root* é mantido em */root* em vez de */home/root*. A razão é simples. O que acontece se o diretório */home* estiver em uma partição diferente do diretório / *e não puder ser montado? O usuário //root* irá querer logar e repara o problema. Se o seu diretório de usuário raíz estiver no sistema de arquivos danificado, será meio difícil para ele conseguir logar-se.

sbin - Programas essenciais que são rodados pelo usuário *root* e durante o processo de boot do sistema são mantidos aqui. Usuários normais não irão rodar programas neste diretório.

tmp - Local de armazenagem temporária. Todos os usuários possuem acesso de leitura e escrita neste diretório.

usr - Este é o maior diretório em um sistema Linux. Tudo o mais vem aqui, programas, documentação, o código-fonte do kernel, e o sistema X Window. Este é o diretório que você mais irá gostar de estar instalando programas.

var - Arquivos de log do sistema, informações de cache, e arquivos protegidos de programas são armazenados aqui. Este é o diretório para mudanças freqüentes de informação.

Você agora deve ter uma boa noção para quais diretórios contém o que no sistema de arquivos. Informações mais detalhadas sobre o layout do sistema de arquivos estão disponíveis nas páginas de manual do hier(7). A próxima seção irá ajudá-lo a encontrar arquivos específicos com facilidade, então você não precisará fazer isso de forma braçal.

4.1.2 Encontrando Arquivos

Agora você sabe o que cada diretório principal contém, mas isso realmente não ajuda em nada a encontrar coisas. Eu creio que você poderia ir olhando através dos diretórios, porém existem

maneiras rápidas de fazer isso. Existem quatro comandos principais de busca de arquivos disponíveis no Slackware.

which

O primeiro é o comando *which*(1). O comando *which* é geralmente usado para achar de forma rápida a localização de um programa. Ele apenas procura seu *PATH* e retorna a primeira instância encontrada e o caminho do diretório do mesmo. Pegue este exemplo: <code> $ which bash /bin/bash </code>

Pelo que você já viu do *bash* ele está no diretório */bin*. Este é um comando de busca muito limitado, já que o mesmo só procura pelo *PATH*.

whereis

O comando *whereis*(1) trabalha de forma similar ao comando *which*, mas pode também procurar por páginas de manual e arquivos fontes. Um comando *whereis* para procurar pelo programa *bash* iria retornar:

<code> $ whereis bash bash: /bin/bash /usr/bin/bash /usr/man/man1/bash.1.gz </code>

Este comando não apenas nos diz onde atualmente o programa está localizado, mas também onde a sua documentação online está armazenada. Ainda assim, este comando é limitado. O que aconteceria se você precisasse procurar por um arquivo de configuração específico? Você não poderia utilizar o comando *which* ou o *whereis* para isso.

find

O comando *find*(1) permite ao usuário procurar o sistema de arquivos com um rico arsenal de opções de busca. Usuários poderiam especificar uma busca com espressões regulares no nome do arquivo, alcance de modificações ou tempo de criação, ou outra propriedade avançada. Por exemplo, para procurar pelo arquivo *xinitrc* padrão em um sistema, o comando a seguir poderia ser utilizado.

<code> $ find / -name xinitrc /var/X11R6/lib/xinit/xinitrc </code>

O comando *find* irá demorar um pouco no processo, desde que ele vai varrer tudo na árvore de diretórios raíz. E se este comando estiver rodando por um usuário normal, irão aparecer várias mensagens erro de permissão negada para os diretórios em que apenas o usuário *root* pode ver. Porém se o comando *find* encontra qualquer arquivo, isto é bom. Se ele pudesse ser pelo menos um pouco mais rápido...

slocate

O comando *slocate*(1) procura no sistema de arquivos inteiro, assim como o comando find pode fazer, porém ele procura em um banco de dados em vez do atual sistema de arquivos. O banco de dados está programado para ser atualizado toda manhã, então você terá de algum modo uma lista atualizada dos arquivos de seu sistema. Você pode rodar manualmente o comando *updatedb*(1) para atualizar o banco de dados slocate (antes de rodar na mão o comando *updatedb*, você precisa primeiro rodar o comando *su* para virar o usuário *root*). Aqui está um exemplo do comando *slocate* em ação:

<code> $ slocate xinitrc # we don't have to go to the root /var/X11R6/lib/xinit/xinitrc /var/X11R6/lib/xinit/xinitrc.fvwm2 /var/X11R6/lib/xinit/xinitrc.openwin /var/X11R6/lib/xinit/xinitrc </code>

Nós conseguimos mais do que estávamos procurando, e de forma rápida também. Com esses comandos, você será capaz de encontrar qualquer coisa que estiver procurando em seu sistema Linux.

4.1.3 O Diretório /etc/rc.d/

Os arquivos de inicialização do sistema são armazenados no diretório */etc/rc.d*. O Slackware utiliza o layout estilo BSD para estes arquivos de inicialização ao contrário dos scripts de inicialização do System V, a qual tende a tornar as mudanças na configuração mais difíceis sem estar utilizando um programa especificamente desenhado para esta função. Nos scripts de inicialização do BSD, cada runlevel é um arquivo rc simples. No System V, cada runlevel é

dado a seu próprio diretório, a qual contém inúmeros scripts de inicialização. Isto provê uma estrutura organizada que é fácil de manter.

Existem várias categorias de arquivos de inicialização. Estes são o sistema de inicialização, runlevel, inicialização de rede, e a compatibilidade com o System V. Pela tradição, nós iremos amontoar todas as outras coisas em uma nova categoria.

Inicialização do Sistema

O primeiro programa a rodar sobre o Slackware, além do kernel Linux, é o *init*(8). Este programa lê o arquivo */etc/inittab*(5) para ver como carrega o sistema. Ele roda o script */etc/rc.d/rc.S* para preparar o sistema antes de ir para o nível de execução desejado. O arquivo *rc.S* habilita sua memória virtual, monta seu sistema de arquivos, limpa certos diretórios de log, inicializa os dispositivos Plug and Play, carrega os módulos do kernel, configura os dispositivos PCMCIA, ativa as portas seriais, e os scripts de inicialização do System V (se encontrado). Obviamente o *rc.S* possui muitos recursos, mas existe alguns scripts no diretório */etc/rc.d* que o *rc.S* irá chamar para completar seu trabalho:

rc.S - Este é o atual script do sistema de inicialização.

rc.modules - Carrega os módulos do kernel. Coisas como sua placa de rede, suporte a PPP, e outras coisas são carregadas aqui. Se este script acha o *rc.netdevice*, correrá tudo bem.

rc.pcmcia - Investiga e configura qualquer dispositivo PCMCIA que você possa ter em seu sistema. Isto é bem útil para usuários de laptops, que provavelmente possuem um modem ou rede PCMCIA.

rc.serial - Configura suas portas seriais rodando os comandos apropriados do *setserial*.

rc.sysvinit - Procura pelos scripts de inicialização do System V para o runlevel desejado e os inicializa. Este assunto é discutido em maiores detalhes abaixo.

Scripts de inicialização do Runlevel

Após o sistema de inicialização estar completo, o comando *init* vai para a inicialização do runlevel. Um runlevel descreve o nível em que a sua máquina irá rodar. Isso soa redundante? Bem, o runlevel comunica ao *init* se você estará aceitando logins de multi-usuários ou apenas de um único usuário, independente se você vai utilizar os serviços de rede ou não, e se você estará utilizando o X Window System ou o comando *agetty*(8) para manipular os logins. Os arquivos abaixo definem os diferentes runlevels no Slackware Linux.

rc.0 - Desliga o sistema (runlevel 0). Por padrão, este runlevel está linkado ao *rc.6*.

rc.4 - Inicialização multi-usuário (runlevel 4), porém no X11 com o KDM, GDM, ou XDM como o gerenciador de login.

rc.6 - Reinicializa o sistema (runlevel 6).

rc.K - Inicializa o sistema em modo mono-usuário (runlevel 1).

rc.M - Modo multi-usuário (runlevels 2 e 3), porém com o login em modo texto padrão. Este é o runlevel padrão no Slackware.

Inicialização de rede

Os Runlevels 2, 3, e 4 irão inicializar os serviços de rede. Os seguintes arquivos são responsáveis pela inicialização do sistema:

rc.inet1 - Criado pelo *netconfig*, este arquivo é responsável pela configuração da atual interface de rede.

rc.inet2 - Roda depois do *rc.inet1* e inicializa os serviços de rede básicos.

rc.atalk - Inicializa os serviços AppleTalk.

rc.httpd - Inicializa o servidor Apache. Como alguns outros scripts, este pode ser utilizado para parar e reiniciar um serviço. O *rc.httpd* toma argumentos de parada, inicio, ou reinicio.

rc.news - Inicializa o servidor de news.

Compatibilidade com System V

A compatibilidade com o System V init foi introduzida no Slackware 7.0. Muitas outras distribuições Linux fazem uso deste estilo, em vez do estilo BSD. Basicamente a cada runlevel é dado um sub-diretório para os scripts do init, do contrário do estilo BSD que dá um script de inicialização para cada runlevel.

O script *rc.sysvinit* irá procurar por todos os scripts de inicialização do System V que você tenha em */etc/rc.d* e inicializá-los, se o runlevel for apropriado. Isto é útil para certos pacotes de software que instalam os scripts de inicialização no System V.

Outros Arquivos

Os scripts descritos abaixo são os outros scripts de inicialização do sistema. Eles são rodados tipicamente de um dos principais scripts descritos acima, então tudo de que você precisa fazer é editar seus conteúdos.

rc.gpm - Inicializa os serviços de habilitação do mouse. Permite que você possa copiar e colar no console do Linux. Ocasionalmente, o gpm irá causar problemas para o mouse quando for usado sobre o X Windows. Se você tiver problemas com o uso do mouse no X, tente tirar a permissão de execução deste arquivo parando o servidor gpm.

rc.font - Carrega as fontes customizáveis para o console.

rc.local - Contém todos os comandos específicos de inicialização do seu sistema. Ele fica vazio depois de uma instalação limpa, esta reservado para administradores locais. Este script é inicializado após todas as outras inicializações terem sido realizadas.

Para habilitar um script, tudo que você precisa fazer é adicionar permissões de execução a ele com o comando *chmod*. Para desabilitar um script, remova a permissão de execução do mesmo. Para maiores informações sobre o comando *chmod*, veja a seção 9.2.

4.2 Selecionando um Kernel

O kernel é a parte do sistema operacional que provê acesso ao hardware, controle de processos, e todo o sistema de controle. O kernel contém suporte para seus dispositivos de hardware, então escolher um kernel para o seu sistema é um passo muito importante.

O Slackware provê mais de uma dúzia de kernels pré-compilados que você poderá utilizar, cada qual com um conjunto padrão de drivers genéricos e alguns drivers específicos. Você pode rodar um dos kernels pré-compilados ou pode construir seu próprio kernel a partir do código-fonte. Seja qualquer uma das opções que for escolhida, você precisará ter certeza que o seu kernel tenha o suporte ao hardware que você possui.

4.2.1 O diretório /kernels no CD-ROM do Slackware

Os kernels pré-compilados estão disponíveis no diretório */kernels* no CD-ROM do Slackware ou no site de FTP no diretório principal do Slackware. Os kernels disponíveis mudam sempre que novas versões são lançadas, então a documentação em cada diretório é o código-fonte do mesmo. O diretório */kernels* possui sub-diretórios para cada kernel disponibilizado. Os sub-diretórios possuem o mesmo nome que o seu disco de boot que o acompanha. Em cada sub-diretório você irá encontrar os seguintes arquivos:

Arquivo	Propósito
System.map	O arquivo de mapas do sistema para este kernel
bzImage	A imagem do kernel atual
config	O arquivo-fonte de configuração para este kernel

Para utilizar um kernel, copie os arquivos *System.map* e *config* para o seu diretório */boot* e copie a imagem do kernel para */boot/vmlinuz*. Rode o */sbin/lilo*(8) para instalar o LILO para este novo kernel, e então reinicie seu sistema. Isso é tudo que há para instalar um novo kernel.

Os kernels que terminam com um .i são kernels IDE. Sendo assim, eles não incluem suporte a SCSI na base do seu kernel. Os kernels que terminam com .s são kernels com suporte a SCSI. Eles incluem todo o suporte a IDE, como nos kernels terminados em .i, além de terem o suporte a SCSI.

4.2.2 Compilando um Kernel a partir do código-fonte

A questão *"Devo compilar um kernel para meu sistema?"* é sempre questionado por usuários novatos. A resposta é sempre talvez. Existem poucas ocasiões onde você irá precisar compilar um kernel específico para o seu sistema. A maioria dos usuários pode utilizar um kernel pré-compilado e seus módulos carregáveis para terem um sistema totalmente funcional. Você irá querer compilar um kernel para o seu sistema apenas se você precisar fazer um upgrade de seu atual kernel para um que não esteja atualmente disponibilizado para o slcakware, ou se você estiver adicionando novas funcionalidades ao seu kernel para dar suporte a um hardware específico, que não esteja presente no código nativo no código-fonte do seu kernel. Qualquer um com um sistema SMP definitivamente irá querer compilar um kernal com suporte a SMP. Também, muitos usuários acham que um kernel customizado rodará. Você achará útil compilar o kernel com otimizações para o processador específico de sua máquina.

Construir seu próprio kernel não é tão difícil assim. O primeiro passo é ter certeza que você tem o código-fonte do kernel instalado no seu sistema. Tenha certeza de ter instalado os pacotes da série K durante o processo de instalação do seu Slackware. Você irá querer ter certeza que possui a série D instalada, especificamente o compilador C, o GNU make, e o GNU binutils. Em geral, é uma boa idéia ter a série D inteira instalada se você planeja fazer qualquer tipo de desenvolvimento. Você também pode baixar a última versão do código-fonte do kernel disponível em http://www.kernel.org/mirrors.

Compilação do Kernel Linux versão 2.4.x

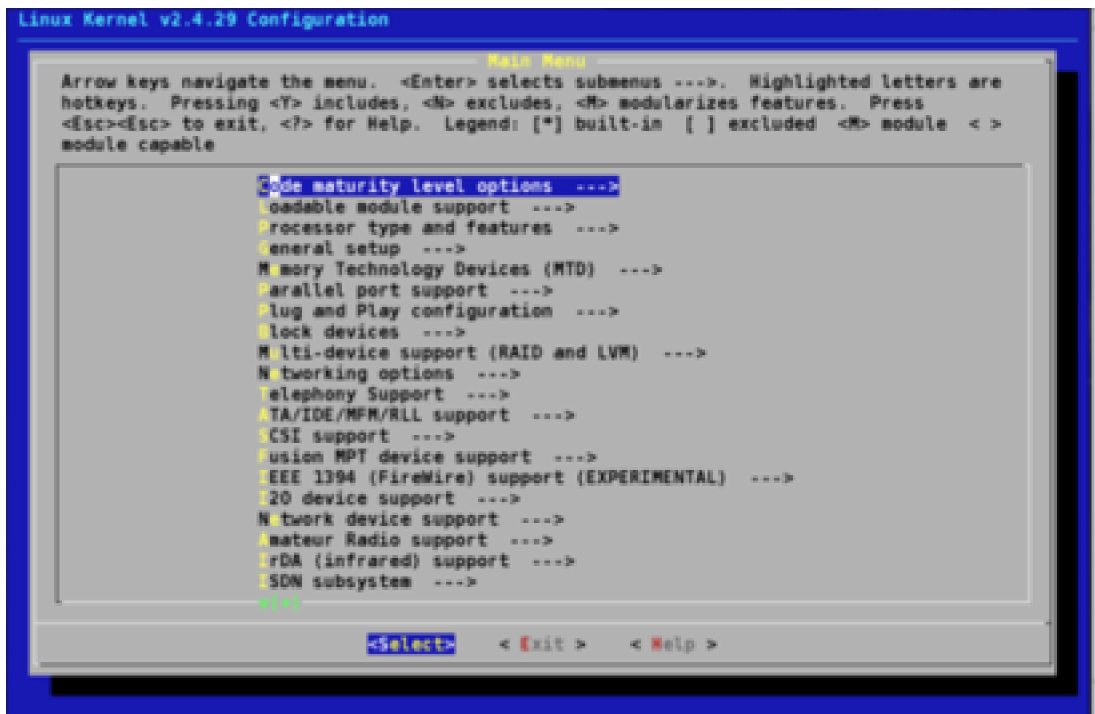
<code> $ su - Password: # cd /usr/src/linux </code>

A primeira etapa é trazer o código-fonte do kernel ao seu estado de base. Nós emitimos este comando para fazer isso (nota, acredito que você vá querer fazer um backup do arquivo *.config* já que esse comando irá apagar ele sem nenhum aviso):

<code> # make mrproper</code>

Agora você pode configurar o kernel para o seu sistema. O kernel atual oferece 3 maneiras de fazer isso. A primeira é o original sistema de perguntas-e-respostas em modo texto. Este fará diversas perguntas e, baseado nas suas respostas, irá construir um arquivo de configuração. O problema deste método é se você errar algo, terá de recomeçar tudo do início. O método que a maior parte dos usuários preferem é o de sistema de menu. Por último, existe uma ferramenta de configuração gráfica do kernel. Escolha uma de sua preferência e digite o comando apropriado:

<code> # make config (baseado em texto, versão P&R (Perguntas e Respostas)) # make menuconfig (menu dirigido, versão baseado em texto) # make xconfig (versão baseado em X, certifique-se que você está primeiramente no X) </code>



menu de configuração do kernel

Os usuários novatos provavelmente irão achar o comando *menuconfig* mais fácil de usar. As telas de ajuda provêm explicações de várias partes do kernel. Após configurar seu kernel, saia do programa de configuração. Ele irá escrever os arquivos de configuração necessários. Agora nós podemos preparar a árvore do código-fonte do kernel para uma construção:

<code> # make dep # make clean </code>

A próxima etapa é compilar o kernel. Primeiro tente emitir o comando *bzImage* abaixo.

<code> # make bzImage </code>

Isso pode demorar um pouco, dependendo da velocidade de sua CPU. Durante o processo de construção, você poderá ver as mensagens geradas pela compilação. Após construir a imagem do kernel, você irá querer construir todas as partes do kernel que você marcou para serem modulares.

<code> # make modules</code>

Você agora pode instalar o kernel e seus módulos que compilou. Para instalar o kernel em um sistema Slackware, os comandos abaixo devem ser digitados:

<code> # mv /boot/vmlinuz /boot/vmlinuz.old # cat arch/i386/boot/bzImage > /vmlinuz # mv /boot/System.map /boot/System.map.old # cp System.map /boot/System.map # make modules_install </code>

Você irá querer editar o arquivo */etc/lilo.conf* e adicionar uma seção para a inicialização de seu kernel antigo em caso do novo kernel não funcionar. Após ter feito isso, execute o comando */sbin/lilo* para instalar o novo bloco de inicialização. Você agora já pode reiniciar com o seu novo kernel.

Kernel Linux Versão 2.6.x

A compilação do kernel 2.6 kernel é levemente diferente do kernel 2.4 ou 2.2, mas é importante que você entenda as diferenças antes de pesquisar. Não é mais necessário rodar os comandos *make dep* e *make clean*. Também, o processo de compilação do kernel não é mais em modo descritivo na série 2.6. Isso resulta em uma construção de processos de fácil entendimento, porém tem seus contras. Se você tiver problemas na construção do kernel, é altamente recomendado que você habilite o modo descritivo de novo. Você pode fazer isso modificando a opção *V=1* na construção. Isso permite a você ter mais informações da saída da compilação que poderá ajudar um desenvolvedor de kernel ou outro amigo entendido a te ajudar na resolução deste problema.

<code> # make bzImage V=1</code>

4.2.3 Utilizando os Módulos do Kernel

Os módulos do kernel é outra definição para os drivers de dispositivo que podem ser inseridos em um kernel em uso. Eles permitem a você extender o suporte ao hardware pelo seu kernel sem precisar usar outro kernel ou mesmo compilar um novo.

Os módulos podem também ser carregados e descarregados a qualquer hora, mesmo se o sistema estiver rodando. Isso torna a atualização de drivers mais fácil para os administradores de sistema. Um novo módulo pode ser compilado, um antigo removido, e o novo ser carregado em seu lugar, tudo isso sem precisar reiniciar sua máquina.

Os módulos são armazenados no diretório */lib/modules/versão do kernel* do seu sistema. Eles podem ser carregados na inicialização do sistema através do arquivo *rc.modules*. Este arquivo é muito bem documentado e oferece exemplos para a maior parte dos componentes de hardware. Para ver a lista dos módulos atualmente ativos, utilize o comando *lsmod*(1):

<code> # lsmod Module Size Used by parport_pc 7220 0 parport 7844 0 [parport_pc] </code>

Você pode ver aqui que eu só tenho o módulo da porta paralela carregado. Para remover um módulo, você usa o comando *rmmod*(1). Os módulos podem ser carregados pelos comandos *modprobe*(1) ou *insmod*(1). *modprobe* é normalmente mais seguro porque ele irá carregar qualquer módulo dos disponíveis que possuam dependências com o carregamento atual.

Vários usuários nunca tiveram que carregar ou descarregar módulos de kernel na mão. Eles utilizam o auto-carregamento do kernel para o gerenciamento de módulos. Por padrão, o Slackware inclui o *kmod* em seu kernel. O *kmod* é uma opção do kernel que habilita o kernel para carregar automaticamente os módulos que são requisitados. Para maiores informações sobre o *kmod* e como ele é configurado, veja o arquivo */usr/src/linux/Documentation/kmod.txt*. Você irá precisar do pacote de código-fonte do kernel, ou baixar o código-fonte do kernel de http://kernel.org.

Maiores informações podem ser encontradas nas páginas de manual para cada um desses comandos, além do arquivo *rc.modules*.

Capítulo 5

Configuração de Rede

5.1 Introdução: netconfig é seu amigo.

Quando você instala inicialmente o Slackware, o programa setup chama o programa *netconfig*. O *netconfig* tem como objetivo realizar para você as seguintes funções:

- Ele te pergunta o nome do seu computador e o nome do domínio sob o qual o seu computador está na rede.
- Ele faz uma breve explanação acerca dos vários tipos de esquemas de endereçamento, cita onde eles devem ser usados, e lhe pergunta qual esquema de endereçamento IP você deseja usar para configurar sua placa de rede:
  - IP estático
  - DHCP
  - Loopback
- Ele então se oferece para detectar alguma placa de rede para configurá-la.

Em geral o *netconfig* irá realizar cerca de 80% do trabalho de configurar sua conexão de rede LAN, se você permitir. Recomenda-se fortemente que você revise seus arquivos de configuração por duas razões:

- Você nunca deve confiar que um programa de configuração irá configurar corretamente seu computador. Se você usa um programa de configuração, você deve revisar a configuração pessoalmente.
- Se você ainda está aprendendo o sistema operacional Linux ou o Slackware, olhar uma configuração funcional pode ser útil. Você vai saber no mínimo qual a aparência da configuração. Isso permitirá que você possa corrigir posteriormente problemas provenientes de configurações erradas realizadas pelo sistema.

5.2 Configuração do Hardware de Rede

Uma vez que seu desejo é conectar sua máquina com o Slackware em algum tipo de rede, a primeira coisa que você precisará é de uma placa de rede compatível com o Linux. Você precisa ter um pouco de cuidado e se certificar de que sua placa é verdadeiramente compatível com o Linux (por favor, consulte o Projeto de Documentação do Linux e/ou a documentação do kernel para informações sobre o status atual da placa de rede que você deseja utilizar). Via de regra você ficará, provavelmente muito surpreso com o número de placas de rede suportadas sob os kernels mais modernos. Tendo dito isso, sugerimos ainda que você consulte algumas das várias listas de hardwares compatíveis com o Linux (como por exemplo The GNU/Linux

Beginners Group Hardware Compatibility Links e The Linux Documentation Project Hardware HOWTO) que estão disponíveis na Internet antes de comprar sua placa. Um pequeno tempo extra gasto em pesquisa podem economizar dias ou até semanas tentando resolver problemas com placas não compatíveis totalmente com o Linux.

Quando você visitar as listas de Hardware Compatível com o Linux disponíveis na Internet, ou quando consultar a documentação do kernel instalado na sua máquina, preste atenção em qual módulo que você irá precisar para suportar sua placa de rede.

5.2.1 Carregando os Módulos de Rede

Os módulos do kernel são carregados durante a inicialização do sistema são carregados pelo arquivo *rc.modules*, */etc/rc.d*, ou pelo carregador automático de módulos do kernel iniciado pelo */etc/rc.d/rc.hotplug*. O arquivo *rc.modules* padrão inclui uma seção de suporte a periféricos de rede. Se você abrir o *rc.modules* e procurar por esta seção, poderá notar que ele verifica inicialmente a existência do arquivo executável *rc.netdevice* em */etc/rc.d/*. Esse script é criado se a tentativa de auto-configuração do periférico de rede executada pelo *setup* obteve sucesso durante a instalação.

Abaixo do bloco do *"if"* está uma lista de periféricos de rede e linhas com o comando modprobe comentadas. Procure o seu periférico e descomente a linha com o modprobe correspondente, e então salve o arquivo. Rodar o *rc.modules* como *root* irá, a partir de agora, carregar o driver do seu periférico de rede (assim como qualquer outro módulo listado e não comentado). Note que alguns módulos (como o do driver ne2000) precisam de parâmetros; tenha certeza que você selecionou a linha correta.

5.2.2 Placas de Rede Local (10/100/1000Base-T and Base-2)

Esse subtítulo abrange todas as placas de rede internas ISA ou PCI. Drivers para essas placas são acessados via módulos carregáveis do kernel conforme dito no parágrafo anterior. É possível que o */sbin/netconfig* tenha encontrado e configurado com sucesso sua placa no seu arquivo *rc.netdevice*. Se isso não ocorreu o problema mais provável deve ser o carregamento do módulo incorreto para a placa em questão (não são conhecidos casos onde para diferentes gerações de um mesmo tipo/família de placas de um mesmo fabricante sejam necessários módulos diferentes). Se você têm certeza que o módulo que você quer carregar é o correto, sua próxima melhor ação seria consultar a documentação do módulo para verificar se o módulo exige ou não algum(uns) parâmetro(s) específico(s) no momento em que o módulo é iniciado.

5.2.3 Modems

Assim como as placas de redes locais, modems podem ter várias opções de barramento suportadas. Até pouco tempo atrás a maioria dos modems eram placas com barramento ISA, de 8 ou 16 bits. Graças aos esforços da Intel e de fabricantes de placas-mãe de todo o mundo, finalmente modems ISA fazem parte do passado, sendo comum atualmente que a maioria dos modems sejam modems externos que se conectam por uma porta serial ou USB, ou então modems internos PCI. Se você deseja trabalhar com seu modem no Linux, é de *VITAL* importância perquisar o modem que você pretende comprar, particularmente se você tem intenção de comprar um modem PCI. Muitos, se não a maioria, dos modems PCI atualmente disponíveis nas prateleiras das lojas são WinModems. WinModems não possuem algumas funções de hardware características dos próprios modems: as funções de responsabilidade desse hardware específico

geralmente são descarregadas sobre a CPU pelo driver do modem e pelo sistema operacional Windows. Isso significa que eles não tem a interface serial padrão que o PPPD estará esperando quando você disca para o seu Provedor de Acesso à Internet.

Se você deseja ter absoluta certeza de que o modem que você está comprando irá trabalhar com o Linux, compre um modem de hardware externo que se conecta à porta serial do seu PC. Com toda certeza esses irão trabalhar melhor e apresentar menos problemas para instalar e dar manutenção, em detrimento de precisarem de energia externa e tenderem a custar mais.

Há vários sites que oferecem drivers e assistência para configuração de periféricos baseados em WinModems. Alguns usuários relatam sucessos na configuração e instalação de drivers para vários Winmodems, incluindo os de chipset Lucent, Conexant e Rockwell. Como os softwares necessários para esses periféricos não fazem parte do Slackware, e variam de driver para driver, nós não iremos entrar em detalhes.

5.2.4 PCMCIA

Como parte da instalação do seu Slackware, você terá a oportunidade de instalar o pacote pcmcia (na série *"A"* de pacotes). Esse pacote contém as aplicações e arquivos de configuração necessários para trabalhar com placas PCMCIA no Slackware. É importante notar que o pacote pcmcia instala apenas o software genérico necessário para trabalhar com placas PCMCIA no Slackware. Ele NÃO instala qualquer driver ou módulo. Os módulos e drivers disponíveis podem ser encontrados no diretório */lib/modules/‘uname -r‘/pcmcia*. Você possivelmente precisará de algumas experiências para encontrar o módulo que irá trabalhar com sua placa de rede. Você precisará editar o arquivo */etc/pcmcia/network.opts* (para uma placa Ethernet) ou o */etc/pcmcia/wireless.opts* (se você tem uma placa de rede sem fio). Como a maioria dos arquivos de configuração do Slackware, esses dois arquivos são muito bem comentados e isso torna fácil a tarefa de determinar que modificações você precisa fazer.

5.3 Configuração TCP/IP

Nesse momento, sua placa de rede deve estar físicamente instalada no seu computador, e os módulos relevantes do kernel devem estar carregados. Você ainda não estará pronto para se comunicar utilizando sua placa de rede, mas informações a respeito do seu periférico de rede podem ser obtidas com um *ifconfig -a*.

```
# ifconfig -a
eth0 Link encap:Ethernet HWaddr 00:A0:CC:3C:60:A4
UP BROADCAST NOTRAILERS RUNNING MULTICAST MTU:1500 Metric:1
RX packets:110081 errors:1 dropped:0 overruns:0 frame:0
TX packets:84931 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:114824506 (109.5 Mb) TX bytes:9337924 (8.9 Mb)
Interrupt:5 Base address:0x8400

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:2234 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:2234 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:168758 (164.8 Kb) TX bytes:168758 (164.8 Kb)
```

Se você digitar */sbin/ifconfig*, sem a opção *-a*, pode ser que você não veja a interface *eth0*, o que indica que sua placa de rede ainda não tem um endereço IP válido ou uma rota.

Uma vez que existem muitas formas diferentes de configurar uma rede, todas elas podem ser reduzidas a dois tipos: Estática e Dinâmica. Redes estáticas são configuradas para que cada nó (nomenclatura geek para objetos que com um endereço IP) sempre tenha o mesmo endereço IP. Redes dinâmicas são configuradas de forma que o endereço IP para cada nó seja controlado por um único servidor, chamado de servidor DHCP.

5.3.1 DHCP

DHCP (Dynamic Host Configuration Protocol ou Protocolo de Configuração Dinâmica de Host), é um método pelo qual um endereço IP pode ser atribuído a um computador quando esse é ligado. Quando o *cliente* é iniciado, ele envia uma requisição na rede local para que um *servidor* DHCP lhe atribua um endereço IP. O servidor DHCP tem um conjunto (ou *escopo*) de endereços IP disponíveis. O servidor irá responder a essa requisição com um endereço IP do seu conjunto, durante um *tempo de alocação*. Uma vez que o tempo de alocação para o endereço IP dado expire, o cliente precisa contactar novamente o servidor e repetir a negociação.

Então, o cliente aceitará o endereço IP enviado pelo servidor e irá configurar a interface que requireu com o endereço IP. Entretanto, há ainda um macete manual que o cliente DHCP usa para negociar o endereço IP que será a ele atribuído. O cliente irá lembrar seu último endereço IP a ele atribuído, e solicitará ao servidor uma re-atribuição do mesmo endereço IP novamente na próxima negociação. Se possível, o servidor irá atender o pedido, mas se não for, um novo endereço será atribuído. Assim, a negociação se parece com o que segue:

```
Cliente: Há algum servidor DHCP disponível na rede local?
Servidor: Sim, há. Aqui estou eu.
Cliente: Eu preciso de um endereço IP.
Servidor: Você pode pegar o 192.168.10.10 por 19200 segundos.
Cliente: Obrigado.

Cliente: Há algun servidor DHCP disponível na rede local?
Servidor: Sim, há. Aqui estou eu.
Cliente: Eu preciso de um endereço IP. Da última vez que conversamos, Eu peguei o 192.16
Posso pegá-lo novamente?
Servidor: Sim, pode (ou Não, não pode: pegue o 192.168.10.12 dessa vez).
Cliente: Obrigado.
```

O cliente DHCP no Linux é o */sbin/dhcpcd*. Se você carregar o */etc/rc.d/rc.inet1* em seu editor de texto favorito, você irá notar que o */sbin/dhcpcd* é chamado mais ou menos no meio do script. Isso irá forçar a o diálogo mostrado acima. O *dhcpcd* irá também controlar a quantidade de tempo de alocação restante para o endereço IP atual, e irá contactar automaticamente o servidor DHCP com uma requisição para renovação da alocação quando necessário. O DHCP pode ainda controlar informações relacionadas, como qual o servidor ntp usar, qual rota pegar, etc.

Configurar DHCP no Slackware é simples. Apenas execute o *netconfig* e selecione DHCP quando oferecido. Se você tem mais de uma placa de rede e não deseja que a *eth0* seja configurada via DHCP, apenas edite o arquivo */etc/rc.d/rc.inet1.conf* e mude a variável relacionada à sua placa de rede para *"YES"*.

5.3.2 IP Estático

Endereços IP estáticos são endereços fixos que só mudam se modificadas manualmente. Isso é usado em qualquer caso em que o administrador não deseje que as informações relacionadas ao IP mudem, por exemplo para servidores internos de uma rede local, qualquer servidor conectado à Internet, e roteadores. Com o endereçamento IP estático, você atribui um endereço IP e esquece :). Outras máquinas saberão que você sempre terá um certo endereço IP e poderão contactar você no endereço de sempre.

/etc/rc.d/rc.inet1.conf

Se você planeja atribuir um endereço IP ao seu novo Slackware, você pode fazê-lo através do script *netconfig*, ou então pode também editar o */etc/rc.d/rc.inet1.conf*. No */etc/rc.d/rc.inet1.conf*, você irá notar:

```
# Primary network interface card (eth0)
    IPADDR[0]=""
    NETMASK[0]=""
    USE_DHCP[0]=""
    DHCP_HOSTNAME[0]=""
```

E, mais abaixo:

GATEWAY=

Nesse caso, nossa tarefa será meramente colocar a informação correta entre as aspas duplas. Essas variáveis são chamadas pelo */etc/rc.d/rc.inet1* no momento do boot para configurar as placas de rede. Para cada placa de rede, apenas insira a informação correta do IP, ou coloque *"YES"* no *USE_DHCP*. O Slackware irá iniciar as interfaces com a informação colocada na ordem em que elas forem encontradas.

A variável *DEFAULT_GW* configura a rota padrão para o Slackware. Todas as comunicações entre seu computador e outros computadores na Internet devem passar através do gateway, no caso de nenhuma outra rota ser especificada para elas (as comunicações). Se você está usando DHCP, em geral você não precisará digitar nada aqui, pois seu servidor DHCP irá especificar qual gateway usar.

/etc/resolv.conf

Muito bem. Então você tem um endereço IP, você tem um gateway padrão, você pode ter dez milhões de reais (nos dê um pouco), mas o que você consegue fazer de bom se você não consegue traduzir nomes para endereços IP? Ninguém merece digitar 72.9.234.112 no navegador para entrar no www.slackbook.org. Além disso, quem, além dos autores conseguiriam memorizar esse endereço IP? Nós precisamos configurar o DNS, mas como? É aí que o */etc/resolv.conf* entra no jogo.

Provavelmente você já tem as opções apropriadas no */etc/resolv.conf*. Se você configurar sua conexão de rede usando o DHCP, o servidor DHCP pode se encarregar de atualizar esse arquivo para você. (Tecnicamente o servidor DHCP apenas diz ao *dhcpcd* o quê colocar aqui, e ele obedece.) No entanto, se você precisa atualizar manualmente sua lista de servidores DNS,

você precisará editar manualmente o */etc/resolv.conf*. Segue um examplo:

```
# cat /etc/resolv.conf
nameserver 192.168.1.254
search lizella.net
```

A primeira linha é simples. A diretiva nameserver nos indica qual servidor DNS procurar. Por necessidade eles sempre são endereços IP. Você pode ter vários listados aqui, conforme seu gosto. O Slackware irá alegremente checar um depois do outro até um deles retornar a busca.

A segunda linha é um pouco mais interessante. A diretiva search nos dá uma lista de domínio de nomes para assumir quando nenhum DNS requisitado for encontrado. Isso lhe permite contactar outra máquina tendo apenas a primeira parte da sua FQDN (Fully Qualified Domain Name, ou Domínio de Nome Totalmente Qualificado). Por exemplo, se *"slackware.com"* estava indicado na sua diretiva search, você pode buscar http://store.slackware.com apenas apontando seu navegador para http://store.

```
# ping -c 1 store
PING store.slackware.com (69.50.233.153): 56 data bytes
64 bytes from 69.50.233.153 : icmp_seq=0 ttl=64 time=0.251 ms
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.251/0.251/0.251 ms
```

/etc/hosts

Agora que nós temos um DNS trabalhando bem, o que podemos fazer se desejamos identificar nosso servidor DNS, ou adicionar uma entrade de DNS em uma máquina que não está no DNS? O Slackware inclui o arquivo salvador */etc/hosts*, que contém uma lista local de nomes DNS e endereços IP para os quais eles apontam.

```
# cat /etc/hosts
127.0.0.1     localhost  locahost.localdomain
192.168.1.101 redtail
172.14.66.32  foobar.slackware.com
```

Nele você pôde notar que o localhost têm como endereço IP relativo o 127.0.0.1 (sempre reservado para o localhost), redtail é apontado para 192.168.1.101, e o foobar.slackware.com para 172.14.66.32.

5.4 PPP

A maioria das pessoas continuam se conectando à Internet através de algum tipo de conexão discada. O método mais comum é o PPP, embora o SLIP seja ainda utilizado ocasionalmente. Configurar o seu sistema para utilizar chamar via PPP um servidor remoto é muito fácil. Nós vamos abordar algumas poucas ferramentas para ajudá-lo(a) a configurar.

5.4.1 pppsetup

O Slackware inclui um programa chamado *pppsetup* para configurar seu sistema para usar sua conta discada. Ele compartilha a aparência e parece muito semelhante ao nosso programa *netconfig*. Para executar o programa, tenha certeza que você está logado como root. Então digite *pppsetup* para executá-lo. Você irá ver uma tela como essa:

O programa apresentará uma série de questões, as quais você precisará responder apropriadamente. Coisas como o seu modem, a string de inicialização do modem, e o número de telefone do seu provedor. Alguns itens terão um valor padrão, que você poderá aceitar na maioria dos casos.

Após rodar o programa, ele criará um programa *ppp-go* e um programa *ppp-off*. Eles são usados para iniciar e parar, respectivamente, a conexão PPP. Os dois programas estão em */usr/sbin* e precisam de privilégios de root para serem executados.

/etc/ppp

Para a maioria dos usuários, rodar o *pppsetup* será o suficiente. Entretanto, podem haver circunstâncias onde você precisará preparar um conjunto de valores a serem usados pelo daemon do PPP. Todas as informações de configuração ficam em */etc/ppp*. Aqui está uma lista do que você pode encontrar para diferentes arquivos:

ip-down	Esse script é executado pelo *pppd* após o encerramento da conexão PPP.
ip-up	Esse script é executado pelo *pppd* quando ocorre sucesso em uma conexão ppp. Coloq
options	Opções gerais para configuração do *pppd*.
options.demand	Opções gerais de configuração do *pppd* quando executado no modo de discagem por d
pppscript	Os comandos enviados para o modem.
pppsetup.txt	Um log dos seus dados de entrada, quando executou o *pppsetup*.

NOTA: A maioria desses arquivos não estarão lá até depois da execução do *pppsetup*.

5.5 Wireless

As redes sem fio ainda são coisas relativamente novas no mundo dos computadores mesmo com o grande crescimento no número de pessoas que compram laptops e esperam encontrar redes por onde passem, sem se preocuparem com o seu velho cabo de par trançado. E essa tendência não parece estar diminuindo o ritmo. Infelizmente, as redes sem fio não são ainda tão bem suportadas no Linux, a exemplo das redes cabeadas tradicionais.

Existem três passos básicos no processo de configuração de uma placa de rede sem fio:

- Suporte ao hardware das placas de rede sem fio
- Configuração da placa para conectar um ponto de acesso sem fio
- Configuração da rede

5.5.1 Suporte ao Hardware

O suporte ao hardware de placas sem fio é dado através do kernel, tanto com um módulo como com suporte direto (built in) no kernel. Geralmente, as placas mais recentes utilizam esse suporte com múdulos. Sendo assim,você pode determinar o módulo aproprioado e carregá-lo

via */etc/rc.d/rc.modules*. O *netconfig* pode não detectar sua placa de rede sem fio. Nesses casos você provavelmente precisará determinar pessoalmente qual o módulo da sua placa. Procure mais informações sobre drivers para o kernel de várias placas de rede sem fio, em http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/.

5.5.2 Configuração da rede sem fio

A vasta maioria desse trabalho é feito pelo *iwconfig*, como sempre você pode ler o manual (com poderes de root, digite man iwconfig no terminal) do *iwconfig* se você precisa de mais informações.

Antes de tudo, você precisará configurar o seu ponto de acesso sem fio. Pontos de acesso sem fio possuem uma pequena variação em suas terminologias, e formas de configuração, então você talvez precise se mexer um pouco para se acostumar ao seu hardware. Em geral, você precisará pelo menos das seguintes informações:

- A ID do domínios, ou o nome da rede (chamado de ESSID pelo *iwconfig*)
- O canal que o ponto de acesso utiliza
- As informações relativas à criptografia, incluindo todas as chaves utilizadas (preferencialmente em hexadecimal)

AVISO: UMA NOTA A RESPEITO DO WEP. O WEP é bastante fraco, mas é muito melhor do que não usar nada. Se você deseja um grau avançado de segurança na sua rede sem fio, você pode estudar VPNs ou IPSec. Ambos estão fora do escopo deste documento. Você pode também configurar seu ponto de acesso para não publicar sua ID de domínio/ESSID. Uma discussão mais completa a respeito de políticas de redes sem fio está além do escopo desta seção, mas uma busca rápida no Google retornará mais do que você esperava aprender.

Uma vez que você tenha coletado as informações acima, e assumindo que você usou o *modprobe* para carregar o driver apropriado no kernel, você pode editar o *rc.wireless.conf* e adicionar suas informações. O arquivo *rc.wireless.conf* é um pouco desorganizado. Um esforço mínimo seria modificar a seção genérica com a sua ESSID, KEY, e CHANNEL, se isso for necessário à sua placa. (Tente não configurar o CHANNEL, e, se funcionar, ótimo! Se não funcionar, configure o CHANNEL de forma apropriada.) Se você for audacioso pode modificar o arquivo de forma que apenas as variáveis necessárias sejam utilizadas. Os nomes das variáveis no *rc.wireless.conf* correspondem aos parâmetros do *iwconfig*. São lidos pelo *rc.wireless* e utilizados nos comandos *iwconfig* apropriados.

Se você tem sua chave em hexadecimal, o que é o ideal, será possível trabalhar de forma bastante confidencial, uma vez que seu ponto de acesso e o *iwconfig* estarão com a chave acordada. Se você tem apenas uma sequência de caracteres, pode ter certeza que o seu ponto de acesso a traduzirá para uma chave em hexadecimal. Nesse caso pode ser necessário algum trabalho para descobrí-la (ou coloque sua chave do ponto de acesso em hexadecimal).

Uma vez que você modificou o *rc.wireless.conf*, execute o *rc.wireless* como *root*, e então rode o *rc.inet1*, novamente como *root*. Você pode testar sua rede sem fio com as ferramentas padrões de testes, como o *ping*, junto com o *iwconfig*.

Se você tem uma interfaces que usem cabo, pode usar, se desejar, o *ifconfig* para desativá-las enquanto estiver testando sua rede sem fio, para ter certeza de não estar havendo nenhuma

confusão de interface. Você pode também desejar testar o efeito das suas mudanças quando da reinicialização do sistema.

Agora que você viu como editar o */etc/rc.d/rc.wireless* para sua rede padrão, vamos dar uma breve olhada no iwconfig para ver como ele trabalha. Isso vai lhe ensinar uma maneira rápida e direta para configurar o wifi para aquelas vezes em que você mesmo procura por sinal de Internet em um café, hotel, aeroporto ou qualquer outro hot spot wifi, desejando ficar online.

O primeiro passo é dizer à sua interface wireless em qual rede entrar. Tenha certeza que você substituiu a *"eth0"* com qualquer outra interface que a sua placa de rede use e trocou a *"mynetwork"* para o essid que você deseja usar. Sim, nós temos certeza que você é esperto o bastante para isso. Depois disso você irá especificar a chave criptográfica (se existir) usada na sua rede sem fio. Finalmente, especifique que canal usar (se necessário).

```
# iwconfig eth0 essid "mynetwork"
# iwconfig eth0 key XXXXXXXXXXXXXXXXXXXXXXXXXX
# iwconfig eth0 channel n
```

Isso encerra nosso assunto sobre redes sem fio.

5.5.3 Configurando a rede

Isso é feito exatamente da mesma maneira como era feito nas redes cabeadas. Simplesmente consulte a primeira parte desse capítulo.

5.6 Sistemas de Arquivo de Rede

Nesse ponto, você já deve ter uma conexão TCP/IP trabalhando na sua rede. Você deve estar conseguindo enviar pings para outros computadores na sua rede interna e, se você configurou corretamente o gateway, deve também conseguir 'pingar' computadores da própria Internet. Como sabemos, o fator básico que nos leva a ligar umcomputador em uma rede é a possibilidade de acessar informações. Enquanto algumas pessoas podem colocar seu computador numa rede apenas para se divertir nela, a maioria deseja compartilhar arquivos e impressoras. Elas desejam poder acessar documentos na Internet ou jogar online. Ter o TCP/IP instalado e functional no seu novo sistema Slackware é um bom inicío para tal, mas apenas com o TCP/IP instalado, as funcionalidades disponíveis serão muito rudimentares. Para compartilhar arquivos, nós precisaremos usar o FTP ou o SCP para tranferí-los para lá e para cá. Não podemos navegar por arquivos em nosso computador Slackware a partir de um vizinho na rede, ou nos ícones do Meus Locais Rede nos computadores com Windows. Da mesma forma, nós também gostaríamos de acessar arquivos em outras máquinas Unix.

Idealmente, nós gostaríamos de poder usar um *sistema de arquivos de rede* que nos permitisse acessar transparentemente nossos arquivos em outros computadores. Os programas que nós usamos para interagir com informações armazenadas em nosso computador não precisam, na verdade, saber em qual computador um dado arquivo está armazenado; eles apenas precisam saber que ele existe e como pegá-lo. É responsabilidade do sistema operacional gerenciar o acesso a arquivos através do sistema de arquivos disponível e dos sistemas de arquivo de rede. Os dois sistemas de arquivo de rede utilizados mais comumente são o SMB (que é implementado pelo Samba) e o NFS.

5.6.1 SMB/Samba/CIFS

O SMB (de Server Message Block) é um descendente do antigo protocolo NetBIOS, que foi inicialmente utilizado pela IBM em seus produtos gerentes de redes locais. A Microsoft também sempre se interessou bastante pelo NetBIOS e seus sucessores (NetBEUI, SMB e CIFS).O projeto Samba existe desde 1991, quando ele foi escrito originalmente para ligar um IBM PC rodando NetBIOS com um servidor Unix. Atualmente o SMB é o método preferido para compartilhamento de arquivos e serviços de impressão sobre uma rede para, virtualmente, toda a população mundial, devido seu suporte ao Windows.

O arquivo de configuração do Samba é o */etc/samba/smb.conf*; um dos mais arquivos de configuração mais bem comentado e documentado que você irá encontrar em qualquer lugar. Exemplos de compartilhamento são configurados para sua visualização, e modificação, se necessário. Se você precisa ter um controle preciso, o man (manual) do smb.conf é indispensável. Uma vez que o Samba é tão bem documentado, conforme mencionamos acima, não iremos reescrever a documentação aqui. Iremos, no entanto, cobrir rapidamente os pontos básicos.

O *smb.conf* é repartido em múltiplas seções: uma seção para compartilhamento, e uma seção global para opções de configuração que válida para todo o escopo do Samba. Algumas opções são aceitas apenas na seção global; outras são válidas apenas fora da seção global. Lembre-se que a seção global pode ser sobreposta por qualquer outra seção. Consulte as páginas do manual (man pages) para mais informações. Você vai se sentir mais confortável se editar seu *smb.conf* de forma a refletir as configurações da sua rede local. Sugerimos a modificação dos itens listados abaixo:

```
[global]
# workgroup = NT-Domain-Name or Workgroup-Name, eg: LINUX2
workgroup = MYGROUP
```

Substitua o nome do grupo de trabalho (workgroup) pelo grupo de trabalho da sua rede, ou pelo nome do domínio que você esteja usando localmente.

```
# server string is the equivalent of the NT Description field
server string = Samba Server<
```

Esse será o nome do seu Slackware, quando mostrado na pasta de computadores vizinhos (ou Meus Locais de Rede).

```
# Security mode. Most people will want user level security. See
# security_level.txt for details. NOTE: To get the behaviour of
# Samba-1.9.18, you'll need to use "security = share".
security = user
```

Muito provavelmente você vai querer implementar o nível de segurança por usuário no seu Slackware.

```
# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba
# documentation.
# Do not enable this option unless you have read those documents
encrypt passwords = yes
```

Se as senhas criptografadas não estão ativadas, você não poderá usar o Samba com os Windows NT 4.0, 2000, XP, e 2003. A princípio, o sistema operacional Windows não requer encriptação para compartilhamento de arquivos.

O SMB é um protocolo de autenticação, o que significa que é necessário a entrada de um usuário e senha corretos, a fim de usar este serviço. Nós avisamos ao servidor samba que os usuários e senhas válidos estão válidos com o comando *smbpasswd*. O *smbpasswd* realiza um par de testes de verificação quando da adição tradicional de usuários, ou da adição de usuários da máquina (o SMB requer que você adicione os nomes dos computadores com NETBIOS como máquinas usuárias, restringindo que computares podem autenticar).

```
Adicionando um usuário ao arquivo /etc/samba/private/smbpasswd.
# smbpasswd -a user

Adicionando um nome de máquina ao arquivo /etc/samba/private/smbpasswd.
# smbpasswd -a -m machine
```

É importante notar que um dado usuário ou nome de máquina precisam existir previamente no arquivo */etc/passwd*. Você pode realizar isso de maneira simples, com o comando *adduser*. Note que quando usamos o comando *adduser* para adicionar um nome de máquina precisamos pós-fixar um sinal de dólar (*'$'*) ao nome da máquina. Entretanto, isso *não* é feito com o *smbpasswd*.O próprio *smbpasswd* se responsabiliza pelo sinal de dólar pós-fixo. Se houver falhas no processo de inserção do nome da máquina com o *adduser* teremos como resultado um erro, quando adicionarmos o nome da máquina no samba.

```
# adduser machine
```

5.6.2 Network File System (NFS)

O NFS (ou Sistema de Arquivos de Redes) foi originalmente escrito pela Sun para sua implementação do Unix, o Solaris. Ao mesmo tempo que ele é significativamente mais fácil de adquirir e rodar, quando comparado ao SMB, também é significativamente menos seguro. A principal vunerabilidade no NFS é que ele é facilmente atingido por técnicas de spoof de usuários ou identificadores de grupo de uma máquina para outra. O NFS é um protocolo que não implementa autenticação. Versões futuras do protocolo NFS estão elaborando melhorias de segurança, mas elas não são comuns até a presente data.

A configuração do NFS é controlada pelo arquivo */etc/exports*. Se você carregar o */etc/exports* padrão em um editor, verá um arquivo em branco com duas linhas comentadas na parte mais superior. Precisaremos adicionar uma linha no arquivo de exportação para cada diretório que quisermos exportar, com uma lista das estações de trabalho clientes para as quais desejemos permitir o acesso aos arquivos. Por hora, se quisermos exportar o diretório */home/diretorio* para a estação de trabalho pc1, podemos simplesmente adicionar a linha:

/home/diretorio pc1(rw) no nosso */etc/exports*. A seguir, você verá um exemplo encontrado na man page (manual) do arquivo *exports*:

```
# sample /etc/exports file
/          master(rw) trusty(rw,no_root_squash)
/projects proj*.local.domain(rw)
/usr       *.local.domain(ro) @trusted(rw)
/home/joe pc001(rw,all_squash,anonuid=150,anongid=100)
/pub       (ro,insecure,all_squash)
```

Como você pode ver, há várias opções disponíveis, e a maioria delas está representada claramente nesse exemplo.

O NFS assume que um dado usuário em uma máquina tem o mesmo user ID (número interno ao sistema operacional, que identifica um usuário) em todas as máquinas na rede. Quando uma consulta de leitura ou escrita de um cliente NFS para um servidor NFS, um UID (user ID) é passada como parte da requisição de leitura/escrita. Esse UID é tratatada da mesma forma de uma requisição de leitura/escrita originada na máquina local. Como você pode ver, se alguém, arbitrariamente, especificar um dado UID quando recursos acessam um sistema remoto, Coisas Ruins (tm) podem e devem ocorrer. Como uma cerca-viva protetora contra isso, cada diretório é montado com a opção *root_squash*. Isso mapeia o UID para qualquer usuário que alegue ser root para um UID diferente, prevenindo então o acesso de root a arquivos ou pastas no diretório exportado. A opção *root_squash* para estar inicialmente habilitada como uma medida de segurança, mas os autores recomendam sua especificação individualmente para todos os diretórios no */etc/exports*.

Você pode também exportar um diretório diretamente utilizando a linha de comando no servidor usando o comando *exportfs*, como segue:

```
# exportfs -o rw,no\_root\_squash pc1:/home/diretorio
```

Essa linha exporta o diretório */home/diretorio* para o computador *"pc1"* e concede acesso de leitura/escrita à máquina de nome pc1. A título de informação o servidor NFS não invocará o *root_squash*, o que significa que qualquer usuário do pc1, com o UID igual a *"0"* (UID do root) terá os mesmo privilégios de root no servidor. A síntaxe parece mesmo um pouco estranha (geralmente quando um diretório é especificado na síntaxe *computador:/diretorio/arquivo*, você está se referindo a um arquivo em um diretório, em um dado computador).

Você pode obter mais informações no manual (man page) do arquivo exports.

Capítulo 6

Configuraração do X

Começando com o Slackware-10.0, o Ambiente de Janelas X é fornecido pelo Xorg. O X é o responsável por fornecer uma interface para o usuário, a qual, diferentemente do Windows ou do MacOS, é independente do sistema operacional.

O Sistema X Window é implementado através de diversos programas executados no ambiente do usuário. Os dois principais componentes são o servidor e o gerenciador de janelas. O servidor fornece as funções de baixo nível para interagir com seu dispositivo de vídeo, sendo dessa forma específico para cada sistema. O gerenciador de janelas está acima do servidor e provê a interface do usuário. A vantagem disso é que você pode ter muitas interfaces de usuário diferentes, simplesmente trocando o gerenciador de janelas que você usa.

Configurar o X pode ser uma tarefa complexa. O motivo dessa complexidade é o grande número de dispostivos de vídeo disponíveis para a arquitetura PC, os quais usam interfaces de programação diferentes. Felizmente, muitas das placas, hoje em dia, suportam os padrões básicos de vídeo, conhecidos como VESA e se sua placa está entre essas, você será capaz de iniciar o X utilizando o comando *startx*, logo após a instalação.

Caso isso não funcione ou você queira obter as vantagens das características de alta-performance de sua placa, como aceleração ou renderização 3-D via hardware, então será necessário reconfigurar o X.

Para configurar o X, será necessário criar o arquivo */etc/X11/xorg.conf*. Esse arquivo contém muitos detalhes sobre seu dispositivo de vídeo, mouse e monitor. É um arquivo de configuração muito complexo, mas, felizmente, existem programas para ajudar na criação de um para você. Iremos mencionar alguns deles aqui.

6.1 xorgconfig

Esta é uma interface simples orientada a menus que é similar ao instalador do Slackware. Ela pede ao servidor X que analize sua placa e então crie o melhor arquivo de configuração inicial que pode ser feito, baseado na informação obtida. O arquivo */etc/X11/xorg.conf* é um bom ponto de partida para a maioria dos sistemas (e deve funcionar sem modificações).

Este é um programa modo-texto de configuração do X, o qual foi desenvolvido para administradores de sistemas experientes. Aqui está um ”passo-a-passo”utilizando o *xorgconfig*. Primeiramente, inicie o programa:

```
# xorgconfig
```

Esta tela apresentará uma tela cheia de informações sobre o *xorgconfig*. Para continuar, pressione ENTER; *xorgconfig* vai propor que verifique se você configurou a variável de ambiente *PATH* corretamente. Ela deve estar correta, assim vá em frente e aperte ENTER.

```
First specify a mouse protocol type. Choose one from the following list:

 1. Auto
 2. SysMouse
 3. MouseSystems
 4. PS/2
 5. Microsoft
 6. Busmouse
 7. IMPS/2
 8. ExplorerPS/2
 9. GlidePointPS/2
10. MouseManPlusPS/2
11. NetMousePS/2
12. NetScrollPS/2
13. ThinkingMousePS/2
14. AceCad

The recommended protocol is Auto. If you have a very old mouse
or don't want OS support or auto detection, and you have a two-button
or three-button serial mouse, it is most likely of type Microsoft.

Enter a protocol number:
```

xorgconfig - Configuração do Mouse

Selecione o seu mouse a partir do menu apresentado. Caso seu mouse serial não esteja listado, selecione o protocolo Microsoft – é o mais comum e provavelmente funcionará. Em seguida, *xorgconfig* irá lhe perguntar se deseja ativar as opções *ChordMiddle* e *Emulate3Buttons*; essas opções estarão descritas em detalhes na tela. Use-as caso o botão do meios de seu mouse não funcionse sob o X, ou caso seu mouse somente tenha dois botões (*Emulate3Buttons* permite a simulação do botão do meio através do pressionamento de ambos os botões simultaneamente). Então, entre com o nome do arquivos de dispositivo de seu mouse; a escolha padrão, */dev/mouse*, deve funcionar desde que a ligação tenha sido escolhida durante a configuração do Slackware. Se você estiver executando o GPM (o servidor de mouse do Linux) no modo de repetição, você poderá configurar o tipo de seu mouse para */dev/gpmdata*, de modo que o X obtém informações de seu mouse através do *gpm*(8). Em alguns casos (especialmente com o "busmice") isso pode funcionar melhor, mas muitos dos usuários não devem fazer isso.

O *xorgconfig* vai lhe perguntar se eseja habilitar o conjunto de teclas especiais. Caso precise disso, diga "y". A maioria dos usuários podem dizer "n-- digite isso se não estiver certo.

```
You must indicate the horizontal sync range of your monitor. You can either
select one of the predefined ranges below that correspond to industry-
standard monitor types, or give a specific range.
It is VERY IMPORTANT that you do not specify a monitor type with a horizontal
sync range that is beyond the capabilities of your monitor. If in doubt,
choose a conservative setting.
    hsync in kHz; monitor type with characteristic modes
 1  31.5; Standard VGA, 640x480 @ 60 Hz
 2  31.5 - 35.1; Super VGA, 800x600 @ 56 Hz
 3  31.5, 35.5; 8514 Compatible, 1024x768 @ 87 Hz interlaced (no 800x600)
 4  31.5, 35.15, 35.5; Super VGA, 1024x768 @ 87 Hz interlaced, 800x600 @ 56 Hz
 5  31.5 - 37.9; Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz
 6  31.5 - 48.5; Non-Interlaced SVGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz
 7  31.5 - 57.0; High Frequency SVGA, 1024x768 @ 70 Hz
 8  31.5 - 64.3; Monitor that can do 1280x1024 @ 60 Hz
 9  31.5 - 79.0; Monitor that can do 1280x1024 @ 74 Hz
10  31.5 - 82.0; Monitor that can do 1280x1024 @ 76 Hz
11  Enter your own horizontal sync range

Enter your choice (1-11):
```

xorgconfig - Sincronia Horizontal

Na próxima seção você informará o intervalo de sincronia horizontal de seu monitor. Para iniciar a configuração de seu monitor, pressione ENTER. Você verá uma lista de tipos de monitores – escolha um deles. Tenha cautela para não exceder a especificação de seu monitor, isso poderá danificá-lo.

```
You must indicate the vertical sync range of your monitor. You can either
select one of the predefined ranges below that correspond to industry-
standard monitor types, or give a specific range. For interlaced modes,
the number that counts is the high one (e.g. 87 Hz rather than 43 Hz).
1  50-70
2  50-90
3  50-100
4  40-150
5  Enter your own vertical sync range

Enter your choice:
```

xorgconfig - Sincronia Vertical

Especifique o intervalo de sincronia vertical de seu monitor (você poderá encontrar isso no manual do monitor). *xorgconfig* irá lhe pedir para entrar com os textos para a identificação do tipo do monitor no arquivo *xorg.conf*. Digite o que quiser nestas três linhas (incluindo nada).

```
 0  * Generic VESA compatible                         -
 1  * Generic VGA compatible                          -
 2  * Unsupported VGA compatible                      -
 3 ** 3DLabs, TI (generic)               [glint]      -
 4 ** 3Dfx (generic)                     [tdfx]       -
 5 ** ATI (generic)                      [ati]        -
 6 ** ATI Radeon (generic)               [radeon]     -
 7 ** ATI Rage 128 based (generic)       [r128]       -
 8 ** Alliance Pro Motion (generic)      [apm]        -
 9 ** Ark Logic (generic)                [ark]        -
10 ** Chips and Technologies (generic)   [chips]      -
11 ** Cirrus Logic (generic)             [cirrus]     -
12 ** Cyrix MediaGX (generic)            [cyrix]      -
13 ** DEC TGA (generic)                  [tgz]        -
14 ** Intel i740 (generic)               [i740]       -
15 ** Intel i810 (generic)               [i810]       -
16 ** Linux framebuffer (generic)        [fbdev]      -
17 ** Matrox Graphics (generic)          [mga]        -

Enter a number to choose the corresponding card definition.
Press enter for the next page, q to continue configuration.
```

xorgconfig - Video Card

Agora você terá a oportunidade de olhar no banco de dados os tipos de placas de vídeo. Provavelmente você vai querer vê-lo, então responda "y"e selecione sua placa na lista exibida; caso você não encontre a placa exata que possui, experimente selecionar uma que use o mesmo chipset (núcleo) e, provávelmente, irá funcionar bem.

Em seguinda, informe ao *xorgconfig* quanto de RAM possui sua placa de vídeo. O *xorgconfig* vai pedir um texto descrição sobre sua placa de vídeo, então, digite sobre a mesma caso deseje.

Você também será perguntado sobre a resolução que deseja usar. Novamente, utilizar os valores padrões pode ser legal para começar, de forma que depois você poderá editar o arquivo */etc/X11/xorg.conf* e reajustar os modos, por exemplo, fazendo que 1024x768 (ou qualquer outro modo que prefira) seja o padrão.

Nesse momento, *xorgconfig* vai perguntar se você quer que o arquivo de configuração seja salvo. Diga que sim e o arquivo de configuração será salvo, completando o processo de configuração. Você poderá iniciar o X com o comando *startx*

6.2 xorgsetup

O segundo caminho para configurar o X é usar o *xorgsetup*, um programa de configuração um – tanto quanto mágico – que vem com o Slackware.

Para executar o *xorgsetup*, entre no sistema como root e digite:

```
# xorgsetup
```

Caso você já tenha um arquivo */etc/X11/xorg.conf* (por já ter configurado o X), você será perguntado se deseja salvar uma cópia de segurança da configuração atual antes de continuar. O arquivo original será renomeado para */etc/X11/xorg.conf.backup*.

6.3 xinitrc

O *xinit*(1) é o programa que inicia o X; ele é chamado pelo *startx*(1), então você não deve ter notado-o (e provavelmente nem precisava). Seu arquivo de configuração, no entanto, determina quais programas (incluindo especialmente os gerenciadores de janela) são executados quando o X inicia. *xinit* primeiramente verifica seu diretório pessoal em busca do arquivo *.xinitrc*. Caso

o arquivo seja encontrado, é executado; caso contrário, */var/X11R6/lib/xinit/xinitrc* (o padrão global do sistema) é utilizado. Segue um exemplo simples do arquivo *xinitrc*:

```
#!/bin/sh
# \$XConsortium: xinitrc.cpp,v 1.4 91/08/22 11:41:34 rws Exp $

userresources=\$HOME/.Xresources
usermodmap=\$HOME/.Xmodmap
sysresources=/usr/X11R6/lib/X11/xinit/.Xresources
sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap

# junta os mapas de teclas padrão

if [ -f $sysresources ]; then
    xrdb -merge $sysresources
fi

if [ -f $sysmodmap ]; then
    xmodmap $sysmodmap
fi

if [ -f $userresources ]; then
    xrdb -merge $userresources
fi

if [ -f $usermodmap ]; then
    xmodmap $usermodmap
fi

\# inicia alguns programas legais


twm &
xclock -geometry 50x50-1+1 &
xterm -geometry 80x50+494+51 &
xterm -geometry 80x20+494-0 &
exec xterm -geometry 80x66+0+0 -name login
```

Todos aqueles blocos ”if”são para mesclar várias configurações de outros arquivos. A parte interessante do arquivo é o final, onde vários programas são executados. Esta sessão do X vai começar com o gerenciador de janelas *twm*(1), um relógio e três terminais. Note que o *exec* antes do último *xterm* sobrepõe a shell em execução, na qual está executando este script *xinitrc*, com o comando *xterm*(1). Assim, quando o usuário fechar o *xterm*, a sessão do X irá terminar.

Para personalizar a incialização do X, copie o arquivo padrão */var/X11R6/lib/xinit/xinitrc* para *˜/.xinitrc* e o edite, substituindo as suas linhas pelo que você desejar. O final do meu é simples:

```
# Inicia o gerenciador de janelas:
exec startkde
```

Note que existem diversos arquivos *xinitrc.** em */var/X11R6/lib/xinit*, os quais correspondem aos vários gerenciadores de janela e interfaces do usuário. Você pode usar qualquer um deles, caso deseje.

6.4 xwmconfig

Durante anos, o Unix foi usado, quase que exclusivamente, como sistema operacional para servidores, com a exceção das poderosas estações de trabalho. Somente os técnicos estavam familiarizados com o uso de um sistema operacional Unix (e suas variações) e as interfaces para o usuário refletiam esse fato. As interfaces gráficas tendiam para ser, de longe, esqueletos, desenhados para executar poucas aplicações gráficas, tais como programas CAD e renderizadores de imagem. Assim, muito do gerenciamento de arquivos e do sistema foi conduzido via linha de comando. Diversos vendedores (Sun Microsystems, Silicaon Graphics, etc) foram vendendo estações que tentavam ter uma aparencia ("look and feel") coesiva, mas a grande variedade de Toolkits (conjunto de componentes para uma interface gráfica) usadas por desenvolvedores levou para a dissolução da uniformidade dos ambientes gráficos. Dessa forma, uma barra de rolagem pode não parecer igual em duas aplicações diferentes; menus podem aparecer em diferentes lugares; programas podem ter diferentes botões e caixas de checagem; havia um grande intervalo de cores e estas eram especificadas no código de cada toolkit. Como os usuários eram primariamente técnicos profissionais, isso não importava muito.

Com o advento de sistemas operacionais gratuitos baseados no Unix e o crescente número e variedade de aplicações gráficas, o X recentemente ganhou uma grande base de usuários no desktop. Muitos usuários, obviamente, estão acostumados com a aparência consistente fornecida pelo Microsoft Windows ou pelo Apple MacOS; a falta de tal consistência em aplicações baseadas no X se tornou uma barreira para sua aceitação. Em resposta, dois projetos de código aberto foram formados: o K Desktop Environment (conhecido como KDE) e o GNU Network Object Model Environment (mais difundido como GNOME). Cada um tem uma variedade de aplicações, que vão desde a barra de tarefas e gerenciadores de arquivo até jogos e suites de escritório, escritas com os mesmos Toolkits da interface e integradas para prover uma área de trabalho uniforme e consistente.

A diferença entre o KDE e GNOME são de longe, grandes. Eles se parecem diferente um do outro por causa da utilização de diferentes Toolkits. O KDE se baseia na biblioteca gráfica QT (da Trolltech AS) enquanto o GNOME usa a GTK, uma biblioteca gráfica originalmente desenvolvida para o GIMP (em português, programa de manipulação de imagens GNU). Como projetos separados, o KDE e o GNOME têm seus próprios desenvolvedores e analistas. O resultado em cada caso, no entanto, tem sido fundamentalmente o mesmo: um ambiente consistente, uma área de trabalho integrada e uma grande coleção de aplicativos. A funcionalidade, usabilidade e detalhes de ambos não é rival de nenhum outro sistema operacional disponível.

A melhor parte, no entando, é que esses ambientes avançados são gratuitos. Isso significa que você pode ter um ou ambos (sim! ao mesmo tempo). A escolha é sua.

Em adição aos ambientes GNOME e KDE, o Slackware inclui uma grande coleção de gerenciadores de janela. Alguns são feitos para emular outros sistemas, outros para personalização e outros para se obter o máximo de desempenho. Existe uma variedade um tanto quanto grande. Obviamente, você pode instalar quantos quiser e, mecher com todos e decidir qual você gostou mais.

Para facilitar a seleção de ambientes de área de trabalho, O Slackware também incluiu um programa chamado *xwmconfig* que pode ser utilizado para selecionar o ambiente de área de trabalho ou gerenciador de janelas. Veja a seguir sua execução:

```
$ xwmconfig
```



Você receberá uma lista de todos os ambientes de área de trabalho e gerenciadores de janelas disponíveis no sistema. Somente selecione da lista o qual deseja. Cada usuário precisará executar este programa, uma vez que diferentes usuários podem usar diferentes ambientes de área de trabalho, e nem todos vão querer o ambiente selecionado por você na instalação.

Então apenas inicie o X e estará pronto.

6.5 xdm

Como Linux se tornou mais e mais usual como um sistema operacional de área de trabalho, muitos usuários acharam interessante o computador ligar diretamente no ambiente gráfico. Para isso, você precisará dizer ao Slackware para inicializar diretamente no X, e atribuir um gerenciador de login gráfico. O Slackware vem com três ferramentas de login gráfico,*xdm*(1), *kdm* e *gdm*(1).

O *xdm* á o gerenciador de login gráfico que vem com o sistema X.org. Ele é constantemente encontrado, mas não possui tantos recursos como os alternativos. *kdm* é o gerenciador de login gráfico do KDE, o Ambiente de área de Trabalho K. Finalmente, *gdm* é o gerenciador de login do GNOME. Quaisquer que sejam a sua escolha vai permitir que você entre no sistema como qualquer usuário e escolha qual ambiente de trabalho deseja usar.

Infelizmente, O Slackware não inclui um bom programa, como o *xwmconfig*, para selecionar qual gerenciador de login será utilizado, então se todos os três estão instalados você pode fazer algumas edições para selecionar o gerenciador de sua preferência. Mas primeiro, vamos discutir como iniciar o sistema no modo gráfico.

A fim de iniciar o X na inicialização, você precisará inicializar no nível de execução (runlevel) 4. Níveis de execução são um modo de dizer ao *init*(8) para fazer algo diferente, quando o sistema operacional é iniciado. Nós fazemos isso alterando o arquivo de configuração do *init*, */etc/inittab*.

```
# Estes são os níveis de execução padrão do Slackware:
#   0 = desligamento
#   1 = modo de usuário singular
#   2 = não utilizado (mas configurado da mesma forma que o nível de execuço 3)
#   3 = modo multi-usuário (nível de execução padrão no Slackware)
#   4 = X11 com KDM/GDM/XDM (gerenciadores de login)
#   5 = inutilizado (mas configura para fazer o mesmo que o nível 3
```

```
#   6 = reinicialização

# nível de execução padrão (Não configurar para 0 ou 6)
id:3:initdefault:
```

A fim de fazer o Slackware inicializar no modo gráfico, somente mudaremos o 3 para o 4.

```
# nível de execução padrão (Não configurar para 0 ou 6)
  id:4:initdefault:
```

Agora o Slackware vai inicializar no nível de execução 4 e executar o */etc/rc.d/rc.4*. Este arquivos inicia o X e chama o gerenciador de login que você escolheu. Agora, como vamos escolher os gerenciadores de login? Existem alguns modos de fazer isso, e vamos explicar eles após em *rc.4*.

```
    # Tenta utilizar o gerenciador de login do GNOME:
  if [ -x /usr/bin/gdm ]; then
    exec /usr/bin/gdm -nodaemon
  fi

  # Não existe?  OK, então tenta utilizar o gerenciador de login do KDE:
  if [ -x /opt/kde/bin/kdm ]; then
    exec /opt/kde/bin/kdm -nodaemon
  fi

  # Se tudo que tivermos for o XDM, acho que ele que vai entrar em ação:
  if [ -x /usr/X11R6/bin/xdm ]; then
    exec /usr/X11R6/bin/xdm -nodaemon
  fi
```

Como você pode ver aqui, *rc.4* primeiro checa se o *gdm* é executável, e, caso seja, executa-o. Em segundo lugar na lista vem o *kdm*, e finalmente o *xdm*. Uma forma de escolher o gerenciador de login é simplesmente removendo os quais você não quer usar utilizando o *removepkg*. Você pode encontrar mais sobre *removepkg* em Gerenciamento de Pacotes do Slackware.

Opcionalmente, você pode remover a permissão de execução dos arquivos que não deseja usar. Nós vamos discutir *chmod* em Estrutura do Sistema de Arquivos.

<code># chmod -x /usr/bin/gdm</code>

Finalmente, você pode somente comentar as linhas dos gerenciadores de login que não quer usar.

```
   # Tenta utilizar o gerenciador de login do GNOME:
  # if [ -x /usr/bin/gdm ]; then
  #   exec /usr/bin/gdm -nodaemon
  # fi

  # Não existe?  OK, entã tenta utilizar o gerenciador de login do KDE:
  if [ -x /opt/kde/bin/kdm ]; then
    exec /opt/kde/bin/kdm -nodaemon
  fi

  # Se tudo que tivermos for o XDM, acho que ele que vai entrar em ação:
  # if [ -x /usr/X11R6/bin/xdm ]; then
  #   exec /usr/X11R6/bin/xdm -nodaemon
  # fi
```

Qualquer linha precedida por um sustenido (#) são consideradas comentários e o interpretador de comandos passa silenciosamente por elas. No entando, mesmo se o *gdm* está instalado e é executável, o interpretador de comandos (nesse caso o *bash*) não vai se importar em checar isso.

Capítulo 7

Processo de Boot

O processo de boot de seu sistema Linux pode tanto ser fácil quanto difícil. Muitos usuários instalam o Slackware em seus computadores e pronto. Eles simplesmente o ligam e já está pronto para ser usado. Para outros, o simples ato de dar boot na máquina pode ser uma dor de cabeça. Para a maioria dos usuários, o LILO funciona melhor. O Slackware inclui o LILO e o LOADLIN para inicializar o Slackware Linux. O LILO funciona a partir de uma partição de um disco rígido, da MBR de um disco rígido, ou de um disquete, o que o torna uma ferramenta versátil. O LOADLIN funciona a partir de uma linha de comando DOS, substituindo o DOS e chamando o Linux.

Outro utilitário popular para inicializar o Linux é o GRUB.

O GRUB não está incluído nem é suportado oficialmente pelo Slackware. O Slackware mantém o padrão *"testado e aprovado"* em tudo o que é incluído na distribuição. Apesar do GRUB funcionar bem e incluir algumas funcionalidades que o LILO não possui, o LILO realiza todas as tarefas essenciais de um sistema de boot de forma confiável e comprovada. Por ser mais recente, o GRUB ainda não conseguiu este legado. Já que não é incluído no Slackware, não vamos discutir sobre ele aqui. Se você quiser usar o GRUB (talvez tenha vindo com outro sistema Linux e você queira usá-lo para dual boot), consulte a documentação do GRUB.

Esta sessão cobre o uso do LILO e do LOADLIN, os dois gerenciadores de boot incluídos no Slackware. Aqui também vemos alguns cenários típicos de dual boot e como configurá-los.

7.1 LILO

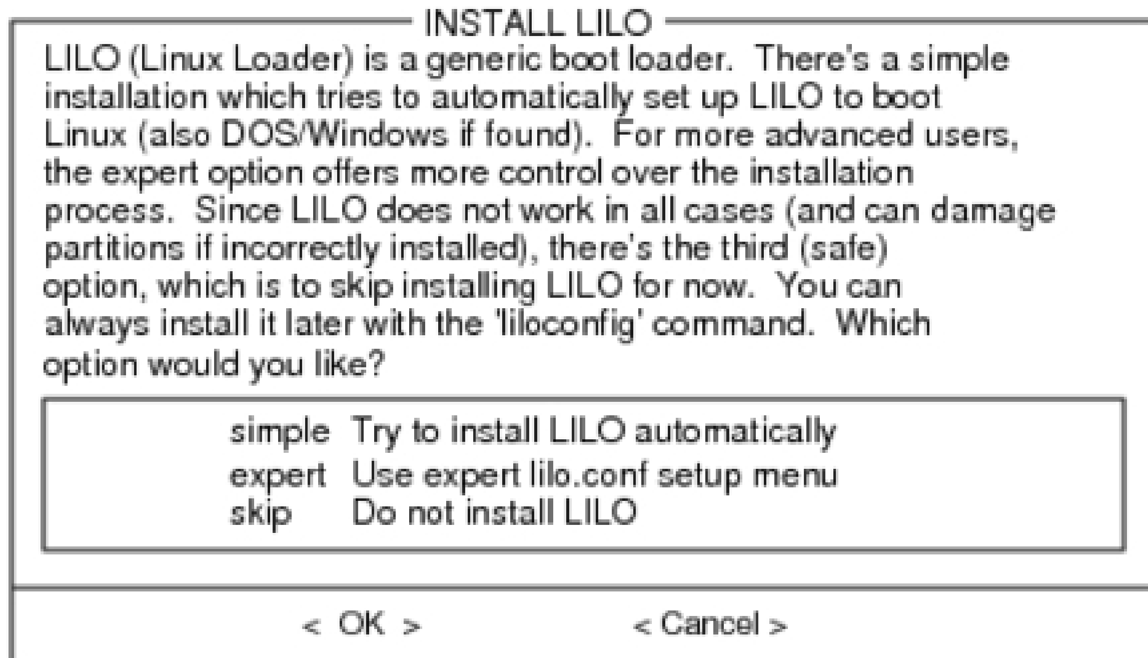
O Linux Loader, ou LILO, é o gerenciador de boot mais usado em sistemas Linux. Ele permite uma extensa gama de configurações pode ser usado facilmente para dar boot em outros sistemas operacionais.

O Slackware Linux possui um utilitário de configuração na forma de menu chamado *liloconfig.* Este utilitário é executado pela primeira vez durante o processo de configuração, mas você pode chamá-lo mais tarde digitando *liloconfig* na linha de comando.

O LILO lê suas configurações a partir do arquivo */etc/lilo.conf*(5) Ele não é lido toda vez que o computador é inicializado, e sim todas as vezes que o LILO é instalado. O LILO deve ser reinstalado no setor de boot sempre que houver uma alteração na configuração. Muitos erros do LILO acontecem quando se faz alterações no arquivo *lilo.conf*, mas se esquece de reexecutar o lilo para instalar essas alterações. O *liloconfig* ajuda a montar o arquivo de configuração para que você possa instalar o LILO em seu sistema. Se preferir editar o */etc/lilo.conf* manualmente,

a reinstalação do LILO envolve simplesmente a digitação de */sbin/lilo* (como *root*) na linha de comando.

Ao chamar pela primeira vez o *liloconfig*, ele será como:



liloconfig

Se esta for a primeira vez que você configura o LILO, escolha a opção simple (simples). Por outro lado, a opção expert (avançado) é mais rápida se você estiver familiarizado com o LILO e o Linux. Selecionando simple, a configuração do LILO é iniciada.

Se o seu kernel estiver compilado com suporte a frame buffer, o *liloconfig* pergunta qual resolução de vídeo você gostaria de usar. Esta resolução também é usada pelo frame buffer do servidor XFree86. Se você não quiser usar o console em um modo de vídeo especial, selecione normal para manter o modo texto padrão de 80x25 em uso.

Na próxima parte da configuração do LILO, selecione onde quer que ele seja instalado. Provavelmente este é o passo mais importante. A lista abaixo explica os locais de instalação:

- Root

Esta opção instala o LILO no início de sua partição raíz. Esta é a maneira mais segura se não houver outros sistemas operacionais em seu computador. Ela assegura que outros gerenciadores de boot não sejam sobreescritos. A desvantagem é que o LILO apenas carrega se o seu drive com o Linux for o primeiro drive em seu sistema. é por este motivo que muitos preferem criar uma pequena partição */boot* como o primeiro driver em seus sistemas. Isto permite que o kernel e o LILO sejam instalados no início do drive, onde o LILO consegue encontrá-los. Algumas versões anteriores do LILO continham uma falha infame conhecida como *”limite do cilindro 1024”*. O LILO não conseguia inicializar kernels em partições além do limite do cilindro 1024. As edições recentes do LILO eliminaram esse problema.

- Disquete

Este método é ainda mais seguro do que o anterior. Ele cria um disquete que você pode usar para dar boot no seu sistema Linux. Isto mantém o gerenciador de boot completamente fora do seu disco rígido. Desse modo, se quiser usar o Slackware, você somente conseguirá iniciar o seu

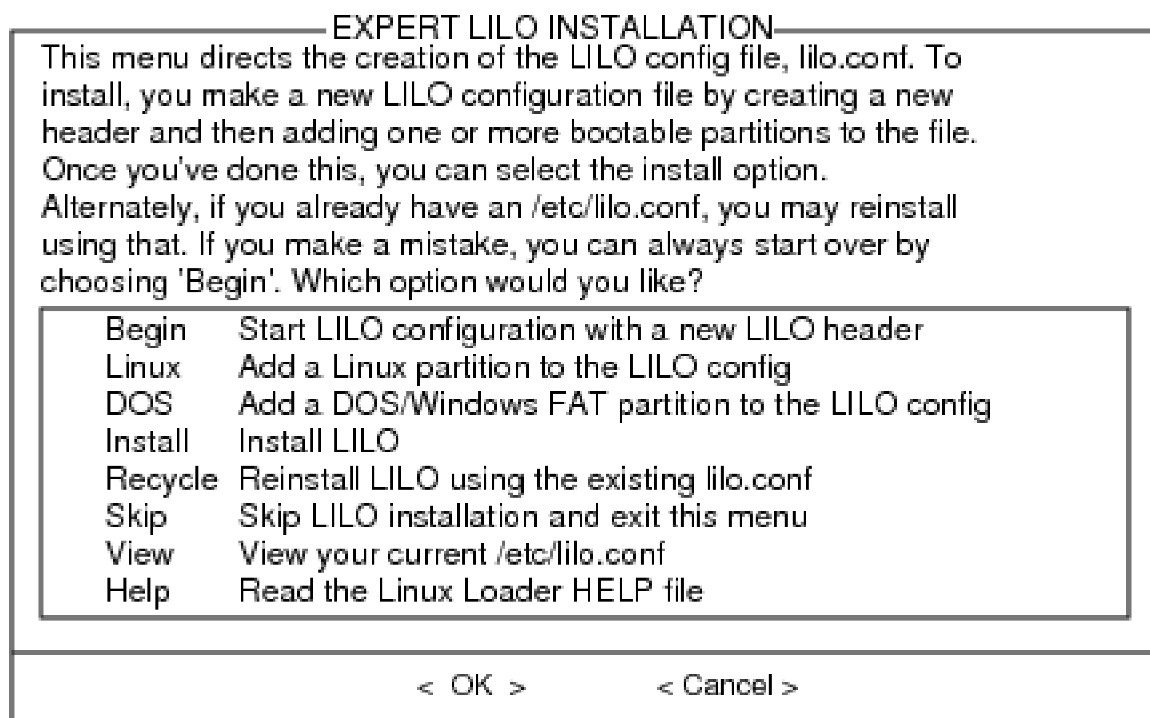
sitema usando este disquete. As falhas deste sistema são óbvias. Os disquetes são conhecidos por serem frágeis, propensos a dar defeitos. Além disso, o gerenciador de boot não está mais contido no computador. Se você perder o seu disquete, terá que fazer outro para inicializar o o seu sistema.

- MBR

Este método é escolhido quando o Slackware é o único sistema operacional em seu computador, ou se o LILO for usado para escolher entre múltiplos sistemas operacionais em seu computador. Este é o método recomendado para a instalação do LILO e funciona em quase qualquer sistema.

AVISO: Esta opção sobreescreve qualquer outro gerenciador de boot que porventura exista na MBR.

Após a seleção do local de instalação, o *liloconfig* cria o arquivo de configuração e instala o LILO. E pronto. Se escolher o modo expert, é apresentado um menu especial. Este menu permite fazer ajustes avançados no arquivo */etc/lilo.conf*, adiconar outros sistemas operacionais no seu menu de boot, e ajustar o LILO para passar parâmetros especiais ao kernel durante o boot. O menu expert se parece com a figura abaixo:



liloconfig - Menu Expert

Qualquer que seja a configuração do seu sistema, é fácil configurar um gerenciador de boot que funcione. O *liloconfig* torna essa tarefa uma moleza.

7.2 LOADLIN

A outra opção para a inicialização que vem com o Slackware Linux é o LOADLIN. O LOADLIN é um executável DOS que pode ser usado para iniciar o Linux a partir de um sistema DOS em execução. Ele requer que o kernel do Linux esteja na partição DOS para que o LOADLIN possa carregá-lo e inicializar o sistema adequadamente.

Durante o processo de boot, o LOADLIN é copiado para o diretório home do usuário root como um arquivo .ZIP. Não há um processo de instalação automática para o LOADLIN. é necessário copiar o kernel do Linux (geralmente */boot/vmlinuz*) e o arquivo LOADLIN do diretório home do usuário root para a partição do DOS.

O LOADLIN é útil quando se deseja fazer um menu de boot em sua partição DOS. Pode-se adicionar um menu ao seu arquivo *AUTOEXEC.BAT* para permitir a escolha entre o Linux e o DOS. Escolhendo o Linux o LOADLIN é executado, assim inicializando o seu sistema Slackware. O exemplo abaixo do arquivo *AUTOEXEC.BAT* para o Windows 95 cria um menu de boot satisfatório:

```
@ECHO OFF
SET PROMPT=$P$G
SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\
CLS
ECHO Selecione o sistema operacional:
ECHO.
ECHO [1] Slackware Linux
ECHO [2] Windows 95
ECHO.
CHOICE /C:12 "Selection? -&gt; "
IF ERRORLEVEL 2 GOTO WIN
IF ERRORLEVEL 1 GOTO LINUX
:WIN
CLS
ECHO Iniciando o Windows 95...
WIN
GOTO END
:LINUX
ECHO Iniciando o Slackware Linux...
CD \LINUX
LOADLIN C:\LINUX\VMLINUZ ROOT=<dispositivo da partição root> RO
GOTO END
:END
```

Especifique a sua partição root com o nome do dispositivo no Linux, tal como */dev/hda2* ou algo semelhante. Você também pode usar o LOADLIN a partir da linha de comando. Simplesmente use-o como no exemplo acima. A documentação do LOADLIN possui muitos exemplos sobre o seu uso.

7.3 Dual Boot

Muitos usuários configuram os seus computadores para inicializar o Slackware Linux e outro sistema operacional (o que se chama dual boot). Descrevemos vários cenários típicos de dual boot abaixo, em caso de haver dificuldades para configurar o seu sistema.

7.3.1 Windows

Provavelmente o cenário mais comum de dual boot é com o MS Windows e o Linux. Há várias maneiras de se configurar a inicialização, mas esta sessão abrangerá duas delas.

Às vezes, ao configurar um sistema com dual boot se cria um plano perfeito em relação a onde tudo vai ficar, mas se confunde a ordem de instalação. é muito importante compreender que os sistemas operacionais devem ser instalados em uma determinada ordem para que o dual boot funcione. O Linux sempre oferece controle sobre o que, se for o caso, é gravado na MBR (Master Boot Record - Registro Principal de Inicialização). Portanto, é aconselhável instalar o Linux por último. O Windows deve ser instalado primeiro, já que ele sempre grava seu gerenciador de boot na MBR, sobreescrevendo as configurações porventura gravadas pelo Linux.

Usando o LILO

A maioria dos usuários preferem usar o LILO para escolher entre o Linux e o Windows. Como foi citado acima, instale primeiramente o Windows, e, então, o Linux.

Digamos que o único drive no seu sistema seja um disco rígido IDE de 40GB. Suponhamos também que você queira deixar a metade do espaço para o Windows e a outra metade para o Linux. Haverá um problema ao tentar iniciar o Linux.

```
20GB   Windows boot (C:)
1GB   Linux root (/)
19GB  Linux /usr (/usr)
```

Você deve separar também um espaço adequado para uma partição de troca (swap) para o Linux. A regra geral é usar o dobro da quantidade de memória RAM. Um sistema com 64MB deve ter 128MB de swap, e assim por diante. O espaço de swap adequado é alvo de muitas discussões inflamadas no IRC e na Usenet. Na verdade não há uma forma *"correta"* de fazer, mas a regra acima deve ser satisfatória.

Com as suas partições prontas, comece a instalação do Windows. Depois que estiver configurado e funcionando, instale o Linux. A instalação do LILO requer atenção especial. Talvez você prefira escolher o modo expert para instalar o LILO.

Inicie uma nova configuração do LILO. Prefira instalar na MBR para que possa ser usado para escolher entre ambos os sistemas operacionais. No menu, adicione a sua partição Linux e a sua partição Windows (ou DOS). Assim que este passo estiver completo, pode instalar o LILO.

Reinicie o seu computador. O LILO deve ser carregado e exibir um menu que permite a escolha entre os sistemas operacionais instalados. Selecione o nome do Sistema Operacional que deseja carregar (esses nomes foram escolhidos na configuração do LILO).

O LILO é um gerenciador de boot altamente configurável. Ele não se limita a iniciar o Linux ou o DOS. Ele pode iniciar praticamente qualquer coisa. As páginas de manual do *lilo*(8) e do *lilo.conf*(5) fornecem informações mais detalhadas.

E se o LILO não funcionar? Há casos onde o LILO não funciona em uma máquina específica. Felizmente, há outra maneira de fazer dual boot entre o Linux e o Windows.

Usando o LOADLIN

Este método pode ser usado se o LILO não funcionar em seu sistema, ou se você não quiser configurar o LILO. Este método também é ideal para o usuário que reinstala o Windows freqüentemente. A cada reinstalação do Windows, ele sobreescreve A MBR, assim destruindo

a instalação do LILO. Com o LOADLIN, você não está sujeito a esse problema. A maior desvantagem é que o LOADLIN somente pode ser usado para iniciar o Linux.

Com o LOADLIN, os sistemas operacionais podem ser instalados em qualquer ordem. Cuidado para não fazer instalações na MBR, você não quer fazer isso. O LOADLIN depende de que a partição do Windows seja inicializável. Então, durante a instalação do Slackware, assegure-se de pular a configuração do LILO.

Após a instalação dos sistemas operacionais, copie o arquivo *loadlin*X*.zip* (onde *X* é o número da versão, tal como *16a*) do diretório home do usuário root para a sua partição Windows. Copie também a imagem do seu kernel para a partição do Windows. Para isso funcionar, é necessário estar no Linux. O exemplo abaixo mostra como fazer isso:

```
# mkdir /win
# mount -t vfat /dev/hda1 /win
# mkdir /win/linux
# cd /root
# cp loadlin* /win/linux
# cp /boot/vmlinuz /win/linux
# cd /win/linuz
# unzip loadlin16a.zip
```

É criado um diretório *C:\LINUX* em sua partição Windows (supondo que seja */dev/hda1*) e tudo que é necessário ao LOADLIN é copiado. Após isso, é necessário reiniciar o Windows para configurar um menu de boot.

De volta ao Windows, abra uma janela do DOS. Primeiro, temos que assegurar que o sistema esteja configurado para não iniciar na interface gráfica.

```
C:\> cd \
C:\> attrib -r -a -s -h MSDOS.SYS
C:\> edit MSDOS.SYS
```

Adicione esta linha ao arquivo:

```
BootGUI=0
```

Agora salve o arquivo e saia do editor. Agora edite o *C:\AUTOEXEC.BAT* para que possamos adicionar um menu de boot. O exemplo abaixo mostra como deve ser um bloco de menu de boot no *AUTOEXEC.BAT*:

```
cls
echo System Boot Menu
echo.
echo 1 - Linux
echo 2 - Windows
echo.
choice /c:12 "Selection? -> "
if errorlevel 2 goto WIN
if errorlevel 1 goto LINUX
:LINUX
cls
echo "Iniciando o Linux..."
```

```
cd \linux
loadlin c:\linux\vmlinuz root=/dev/hda2 ro
goto END
:WIN
cls
echo "Iniciando o Windows..."
win
goto END
:END
```

A linha principal é a que chama o LOADLIN. Dizemos qual kernel carregar, a partição root do Linux, e que inicialmente queremos que seja montada somente para leitura.

As ferramentas necessárias para esse método são fornecidas com o Slackware Linux. Há vários outros gerenciadores de boot no mercado, mas esses devem funcionar ma maioria das configurações de dual boot.

Esta é a situação de dual boot menos comum. Nos velhos tempos do o LILO não conseguia iniciar o Windows NT, o que requeria que os usuários do Linux alterassem a NTLDR, que apresentava muito mais problemas do que o dual boot entre o Windows 9x e o Linux. Esteja ciente de que as instruções abaixo estão ultrapassadas. O LILO já consegue iniciar o Windows NT/2000/XP/2003 há vários anos. Contudo, se você estiver usando uma máquina antiga, este procedimento pode ser necessário.

- Instale o Windows NT
- Instale o Linux, assegurando-se de que o LILO seja instalado no superbloco da partição do Linux
- Coloque os primeiros 512 bytes da partição root do Linux na partição do Windows NT
- Edite o *C:\BOOT.INI* sob o Windows NT para adicionar uma opção para o Linux

A instalação do Windows NT deve ser bem direta, assim como deve ser a instalação do Linux. A partir desse ponto, fica mais complicado. Pegar os primeiros 512 bytes da partição do Linux é mais fácil do que parece. Para isso, você precisa estar no Linux. Supondo que a sua partição Linux seja */dev/hda2*, digite o comando:

```
# dd if=/dev/hda2 of=/tmp/bootsect.lnx bs=1 count=512
```

Só isso. Agora você precisa copiar o bootsect.lnx para a partição do Windows NT. Aqui ocorre outro problema. O Linux não possui suporte estável para gravação em sistemas NTFS. Se você tiver instalado o Windows NT e formatado o seu drive como NTFS, será necessário copiar esse arquivo para um disquete com FAT e então ler a partir dele no Windows NT. Se você tiver formatado o drive do Windows NT como FAT, pode simplesmente montá-lo no Linux e copiar o arquivo. De qualquer modo, copie o */tmp/bootsect.lnx* do Linux para *C:\BOOTSECT.LNX* no Windows NT.

O último passo é adicionar uma opção no menu de boot do Windows NT. No Windows NT, abra um terminal.

```
C:\WINNT> cd \
C:\> attrib -r -a -s -h boot.ini
C:\> edit boot.ini
```

Adicione esta linha ao final do arquivo:

```
C:\bootsect.lnx="Slackware Linux"
```

Salve as alterações e saia do editor. Ao reiniciar o Windows NT, aparecerá uma opção para o Linux no menu. Selecione-a para iniciar o Linux.

7.3.2 Linux

Sim, isso realmente é feito. Definitivamente, este é o cenário de dual boot mais simples. Pode-se simplesmente usar o LILO e adicionar mais entradas no arquivo */etc/lilo.conf*. Isso é tudo.

Capítulo 8

O Shell

Em um ambiente gráfico, a interface é apresentada por um programa que cria janelas, barras de rolagem, menus, etc. Em um ambiente de linha de comando, a interface com o usuário é apresentada por um shell, que interpreta os comandos e geralmente torna as coisas úteis. Imediatamente após entrar no sistema (o que é visto neste capítulo), os usuários são colocados em um shell e podem fazer o seu trabalho. Este capítulo serve como uma introdução ao shell, e ao shell mais comum entre os usuários do Linux - o Bourne Again Shell (bash). Para maiores informações sobre qualquer tópico deste capítulo, veja a página de manual do *bash*(1).

8.1 Os Usuários

8.1.1 Entrando no Sistema

Então você iniciou o Linux, e está olhando para algo que se parece com isso:

```
Welcome to Linux 2.4.18
Last login: Wed Jan 1 15:59:14 -0500 2005 on tty6.
darkstar login:
```

Hã... ninguém disse nada sobre um login. E o que é darkstar? Não se preocupe; provavelmente você não abriu acidentalmente um link de comunicação espacial com a lua artificial do Império. (Infelizmente, o protocolo de link de comunicação espacial ainda não tem suporte no kernel do Linux. Talvez a série 2.8 do kernel finalmente tenha suporte a esse protocolo tão esperado.) Não, darkstar é apenas o nome de um dos nossos computadores, e seu nome aparece por padrão. Se você tiver especificado um nome para o seu computador durante a configuração, ele deve ser exibido em vez de darkstar.

Quanto ao login... Se for a sua primeira vez, você deve entrar como *root*. Será pedida uma senha; se você tiver criado uma durante o processo de configuração, é essa senha que será pedida. Se não, simplesmente pressione enter. Isso aí - você entrou!

8.1.2 Root: O Super Usuário

Ok, quem ou *o que* é o *root*? E o que está fazendo com uma conta em *meu* sistema?

Bem, no mundo Unix e dos sistemas operacionais semelhantes (como o Linux), há usuários e usuários. Veremos maiores detalhes depois, mas o que importa por enquanto é saber que o *root* é o maior usuário de todos; o *root* é onipotente e onisciente, e *ninguém* desobedece o *root*. Isso simplesmente não é permitido. O *root* é o que chamamos de *"super usuário"*, e realmente ele é. E o que é melhor, o *root* é *você*.

Legal, hein?

Se você estiver em dúvida: sim, isso é muito legal. O problema é, contudo, que o root pode por sua natureza, pode quebrar tudo o que desejar. Você pode querer ir direto para a seção 12.1.1 e ver como adicionar um usuário; e, então, entre no sistema e continue daqui. A sabedoria popular diz que é melhor só se tornar o super usuário quando for absolutamente necessário, de forma a minimizar a possibilidade de estragar algo acidentalmente.

A propósito, não há problemas se você decidir que quer se tornar o root enquanto está no sistema como outro usuário. É só usar o comando *su*(1). Você terá que informar a senha de *root* e se tornará o *root* até que dê o comando *exit* ou *logout*. Você também pode se tornar qualquer outro usuário usando o *su*, desde que saiba a sua senha: *su logan*, por exemplo, permitiria que você fosse eu.

NOTA: O *root* pode dar o *su* para qualquer usuário, sem precisar de senha.

8.2 A Linha de Comando

8.2.1 Executando Programas

Não há muito o que fazer sem executar um programa; deve dar para apoiar algo com o seu computador ou manter uma porta aberta, e alguns fazem um zumbido maravilhoso quando estão ligados, mas não é para isso que servem. E acho que podemos concordar que ser usado como batente para portas não é o que trouxe popularidade aos computadores pessoais.

Então, lembra-se de que quase tudo no Linux é um arquivo? Bem, isso também vale para programas. Cada comando que você executa (que não é função nativa do shell) reside em um arquivo em algum lugar. Você executa um programa simplesmente especificando o caminho completo até ele.

Por exemplo, lembra-se do comando *su* da seção acima? Bem, na verdade ele está no diretório */bin*: */bin/su* o executaria perfeitamente.

Por que, então, digitar apenas *su* funciona? Afinal, você não disse que ele estava em */bin*. Poderia também estar em */usr/local/share*, certo? Como o sistema *sabia*? A resposta está na variável de ambiente *PATH*; a maioria dos shells possui o *PATH* ou algo muito parecido com o *PATH*. Essa variável basicamente contém uma lista de diretórios, para procurar os programas que você tenta executar. Então, quando você executou *su*, o seu shell percorreu sua lista de diretórios, procurando em cada um o arquivo executável chamado *su* que pudesse executar; assim que encontrou algum, ele o executou. Isto acontece sempre que você executa um programa sem especificar o caminho completo para ele; se der um erro de *"Command not found"* (comando não encontrado), isso significa que o programa que tentou executar não está em seu *PATH*. (É claro que isso seria verdadeiro se o programa não existisse...) Discutiremos as variáveis de ambiente com maiores detalhes em variáveis de ambiente.

Lembre-se, também, que o *"."* é um atalho para o diretório atual, então se você estivesse em */bin*, *./su* funcionaria como um caminho completo explícito.

8.2.2 Substituição de Meta-Caracteres

Quase todo shell reconhece alguns caracteres como substitutos ou abreviações que significam que qualquer caracter encaixa aqui. Estes caracteres são chamados de meta-caracteres; os mais comuns são *** e *?*. Por convenção, o *?* geralmente substitui qualquer caractere único. Por exemplo, suponha que esteja em um diretório com três arquivos: *ex1.txt*, *ex2.txt*, e *ex3.txt*. Você quer copiar todos estes arquivos (usando o comando *cp* visto na seção 10.5.1) para outro diretório, digamos */tmp*. Bem, digitar *cp ex1.txt ex2.txt ex3.txt /tmp* é trabalhoso demais. É muito mais fácil digitar *cp ex?.txt /tmp*; o *?* combina com qualquer um dos caracteres *"1"*, *"2"*, e *"3"*, e cada um, por sua vez, será substituído.

O que disse? Isto *ainda* dá muito trabalho? Você está certo. Isto é assustador; temos leis trabalhistas para nos proteger deste tipo de coisa. Felizmente, também temos o ***. Como já foi mencionado, o *** substitui *"qualquer número de caracteres"*, inclusive 0. Então se aqueles três arquivos fossem os únicos no diretório, poderíamos simplesmente dizer *cp * /tmp* e pegar todos com um só golpe. Suponha, contudo, que também exista um arquivo chamado *ex.txt* e um chamado *hejaz.txt*. Queremos copiar o *ex.txt*, mas não o *hejaz.txt*; O *cp ex* /tmp* faz isso para nós.

O *cp ex?.txt /tmp*, somente iria, é claro, pegar os nossos três arquivos originais; não há um caractere em *ex.txt* para combinar com o *?*, então ele ficaria de fora.

Outro meta-caractere comum é o par de colchetes *[ ]*. Quaisquer caracteres dentro dos colchetes substituem de acordo com o padrão informado em *[ ]* para encontrar combinações. Parece confuso? Não é tão ruím. Suponha, por exemplo, que um diretório contém os 8 seguintes arquivos: *a1*, *a2*, *a3*, *a4*, *aA*, *aB*, *aC*, e *aD* . Queremos encontrar apenas os arquivos que terminam com números; o *[ ]* fará isso para nós.

```
$ ls a[1-4] a1 a2 a3 a4
```

Mas o que realmente queremos é apenas os arquivos *a1*, *a2* e *a4*? no exemplo anterior usamos - significando todos os valores entre 1 e 4. Também podemos separar entradas individuais usando vírgulas.

```
$ ls a[1,2,4] a1 a2 a4
```

Sei que agora você está pensando: - *Bem, e quanto as letras?* O Linux distingue entre maiúsculas e minúsculas, o que significa que *a* é diferente de *A* e que só são relacionados em sua mente. As letras maiúsculas sempre vêm antes das letras minúsculas, então *A* e *B* vêm antes de *a* e *b*. Dando continuidade ao nosso exemplo anterior, se quiséssemos os arquivos *a1* e *A1*, podemos encontrá-los rapidamente com o *[ ]*.

```
$ ls [A,a]1 A1 a1
```

Observe que se incluíssemos um hífen em vez de uma vírgula, obteríamos resultados incorretos.

```
$ ls [A-a]1 A1 B1 C1 D1 a1
```

Também pode combinar hífen e vírgula.

```
$ ls [A,a-d]1 A1 a1 b1 c1 d1
```

8.2.3 Redirecionamento de Entrada e Saída e Uso do Pipe

(Lá vem algo legal.)

```
$ ps > blargh
```

Sabe o que é isso? Estou executando o *ps* para ver quais processos estão em execução; o *ps* está na seção 11.3. Essa não é a parte legal. A parte legal é > *blargh*, que significa, a grosso modo, pegue a saída do *ps* e mande para um arquivo chamado *blargh*. Mas espere, vai ficar mais legal.

```
$ ps | less
```

Este pega a saída do *ps* e a manda através do *less*, então posso subir ou descer a página conforme a minha vontade.

```
$ ps >> blargh
```

Este é o terceiro redirecionador mais usado; ele faz o mesmo que o ”>”, exceto pelo fato de que o ”>>” adiciona a saída do *ps* ao final do arquivo *blargh*, se este arquivo existir. Se não, assim como o ”>”, ele será criado. (O ”>” substitui o conteúdo atual do arquivo *blargh*.)

Também há o operador ”<”, que significa pegue a entrada a partir do saída, mas não é usado tão freqüentemente.

```
$ fromdos < dosfile.txt > unixfile.txt
```

O redirecionamento fica mesmo divertido quando começa a ser combinado:

```
$ ps | tac >> blargh
```

Isso executa o *ps*, reverte as linhas de sua saída, e as adiciona ao arquivo *blargh*. Você pode empilhar quantos redirecionadores desejar; apenas tenha o cuidado de se lembrar de que eles são interpretados da esquerda para a direita.

Veja a página de manual do *bash*(1) para encontrar informações mais detalhadas sobre redirecionamento.

8.3 O Bourne Again Shell (bash)

8.3.1 Variáveis de Ambiente

Um sistema Linux é uma fera complexa, e há muito para se lembrar, muitos pequenos detalhes que fazem diferença nas suas interações normais com vários programas (alguns dos quais você não deve nem ter ouvido falar). Ninguém precisa passar um monte de opções para cada programa que é executado, dizendo a ele que tipo de terminal está sendo usado, o nome da máquina, como o seu prompt deve ser...

Exemplo: Listando as Variáveis de Ambiente com o *set*

```
$ set
PATH=/usr/local/lib/qt/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/usr/openwin/bin:
:/usr/local/ssh2/bin:/usr/local/ssh1/bin:/usr/share/texmf/bin:/usr/local/sbin:/usr/sbin:
PIPESTATUS=([0]="0")
PPID=4978
PS1='\h:\w\$ '
PS2='> '
PS4='+ '
PWD=/home/logan
QTDIR=/usr/local/lib/qt
REMOTEHOST=ninja.tdn
SHELL=/bin/bash
```

Então, como um mecanismo de ajuda, os usuários contam com o que se chama de ambiente. O ambiente define as condições nas quais os programas são executados, e algumas destas definições são variáveis; o usuário pode alterar e brincar com elas, de forma que só acontece em um sistema Linux. Praticamente qualquer shell possui variáveis de ambiente (se não, provavelmente não é um shell muito útil). Daremos uma visão geral dos comandos do bash para a manipulação das suas variáveis de ambiente.

O *set* sozinho exibe todas as variáveis de ambiente definidas atualmente, assim como seus valores. Como a maioria das funções nativas do *bash*, também pode fazer várias outras coisas (com parâmentros); deixaremos a página de manual do *bash*(1) cobrir esses detalhes, entretanto. O Exemplo acima mostra um trecho da saída de um comando *set* executado em um dos computadores do autor. Observe nesse exemplo a variável *PATH* que foi discutida acima. Os programas de cada um destes diretórios podem ser executados simplesmente digitando-se o nome do arquivo.

```
$ unset VARIAVEL
```

O comando *unset* remove as variáveis que você indicar, apagando tanto a variável quanto o seu conteúdo; O *bash* esquece que a variável já existiu. (Não se preocupe. A não ser que seja algo que você definiu explicitamente naquela sessão do shell, provavelmente será redefinida em uma outra sessão.)

```
$ export \textbf{VARIAVEL=algum\_valor}
```

O comando *export* é realmente prático. Usando-o, você dá à variável de ambiente *VARIAVEL* o valor *"algum_valor"*; se *VARIAVEL* não existia, passa a existir. Se *VARIAVEL* já possuir um valor, bem, já era. Isso não é tão bom se você estiver apenas tentando adicionar um diretório à sua variável *PATH*. Neste caso, você provavelmente quer fazer alguma coisa como:

```
$ export PATH=\$PATH:\textbf{/um/novo/diretorio}
```

Observe o uso de *$PATH* aqui: quando se quer que o *bash* interprete uma variável (substituí-la por seu valor), adicione um *$* ao início do nome da variável. Por exemplo, *echo $PATH* exibe o valor de *PATH*, no meu caso:

```
$ echo \$PATH
/usr/local/lib/qt/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:
/usr/openwin/bin:/usr/games:.:/usr/local/ssh2/bin:/usr/local/ssh1/bin:
/usr/share/texmf/bin:/usr/local/sbin:/usr/sbin:/home/logan/bin
```

8.3.2 Completando Linhas de Comando com o TAB

(Aqui vai outra coisa legal.)

- Usar uma interface de linha de comando significa digitar muito.
- Digitar dá trabalho.
- Ninguém gosta de trabalho.

  A partir dos itens 2 e 3, podemos determinar que (4) ninguém gosta de digitar. Felizmente, o *bash* nos salva do item (5) (Ninguém gosta de uma interface de linha de comando).

  Você está perguntando como o *bash* consegue esta proeza maravilhosa? Além da expansão de meta-caracteres discutido acima, o *bash* apresenta a capacidade de completar linhas de comando com o TAB.

  A capacidade de completar linhas de comando com o TAB funciona assim: Você está digitando o nome de um arquivo. Talvez esteja em seu *PATH*, talvez você esteja digitando explicitamente. Tudo que se precisa é digitar o suficiente do nome do arquivo para identificá-lo de forma única. Então pressione a tecla TAB. O *bash* descobre o que você quer e termina a digitação para você!

  Veja o exemplo. O */usr/src* contém dois subdiretórios: */usr/src/linux* e */usr/src/sendmail*. Quero ver o que está em */usr/src/linux*. Então simplesmente digito *ls /usr/src/l*, pressiono a tecla TAB, e o *bash* me dá o comando *ls /usr/src/linux*.

  Agora, suponha que existam dois diretórios */usr/src/linux* e */usr/src/linux-old*; Se eu digitar */usr/src/l* e pressionar TAB, o *bash* preenche tanto quanto consegue, e vou ter */usr/src/linux*. Posso parar aqui, ou posso pressionar TAB novamente, e o *bash* exibe uma relação de diretórios que combinam com o que já digitei até então.

  Quer dizer, menos digitação (e então, as pessoas podem gostar de interfaces de linha de comando). Eu disse que era legal.

8.4 Terminais Virtuais

Você está no meio de um trabalho e decide que precisa fazer outra coisa. Você poderia simplesmente parar o que está fazendo e alternar as tarefas, mas este sistema é multi-usuário, não é? E você pode entrar no sistema quantas vezes quiser simultaneamente, não pode? Então por que teria que fazer uma coisa de cada vez?

Não precisa. Nem todos nós podemos ter múltiplos teclados, mouse e monitores para cada máquina; provavelmente nem queremos ter. Obviamente, hardware não é a solução. Sobra para o software, e o Linux se dá bem nesse caso, oferecendo *"terminais virtuais"*, ou *"VTs"*.

Pressionando **Alt** e uma tecla de função, você pode alternar entre os terminais virtuais; cada tecla de função corresponde a um deles. Por padrão, o Slackware tem logins em 6 terminais virtuais. **Alt+F2** te leva ao segundo, **Alt+F3** ao terceiro, etc.

O restante das teclas de função são reservadas para sessões do X. Cada sessão do X usa seu próprio VT, começando pelo sétimo (**Alt+F7**) e assim por diante. No X, a combinação

Alt+Tecla de Função é substituída por **Ctrl+Alt+Tecla de Função**; assim, se estiver no X e quiser voltar a um console de texto (sem sair de sua sessão no X), **Ctrl+Alt+F3** te levará ao terceiro terminal virtual. (**Alt+F7** te leva de volta à primeira sessão do X.)

8.4.1 Screen

E quanto às situações onde não há terminais virtuais? E então? Felizmente, o Slackware inclui um belo gerenciador de tela oportunamente chamado de *screen*. O *screen* é um emulador de terminal que possui capacidades semelhantes às de um terminal virtual. Ao executar o *screen*, uma breve apresentação aparece, então um terminal abre. Diferentemente dos terminais virtuais padrão, o *screen* possui seus próprios comandos. Todos os comandos do *screen* são precedidos por **Crtl+A**. Por exemplo, **Ctrl+A+C** cria uma nova sessão do terminal. **Ctrl+A+N** alterna para o próximo terminal. **Ctrl+A+P** alterna para o terminal anterior.

O *screen* também permite sair e retornar a sessões do *screen*, o que é especialmente útil em sessões remotas por *ssh* e *telnet*, (maiores detalhes abaixo). **Ctrl+A+D** sai da tela atualmente em execução. O comando *screen -r* relaciona todas as sessões atuais do screen às quais você pode retornar.

```
$ screen -r
There are several suitable screens on:
     1212.pts-1.redtail(Detached)
     1195.pts-1.redtail(Detached)
     1225.pts-1.redtail(Detached)
     17146.pts-1.sanctuary   (Dead ???)
Remove dead screens with 'screen -wipe'.
Type "screen [-d] -r [pid.]tty.host" to resume one of them.
```

O comando *screen -r 1212* retorna à primeira sessão listada. Mencionei anteriormente que isso era útil para sessões remotas. Se eu tivesse logado em um servidor Slackware remoto por *ssh*, e a minha conexão caísse pro algum motivo como falta de energia local, o que quer que eu estivesse fazendo naquele momento seria perdido instantaneamente, o que pode ser terrível para o seu servidor. Usar o *screen* evita isso, desconectando a minha sessão se a minha conexão cair. Assim que a minha conexão for restaurada, posso retornar à minha sessão do screen e reiniciar exatamente de onde parei.

Capítulo 9

Estrutura do Sistema de Arquivos

Nós já discutimos previamente sobre a estrutura de diretórios no Slackware Linux. A partir deste ponto, você está apto a encontrar arquivos e diretórios que precisar. Porém, há mais sobre sistemas de arquivos que apenas a estrutura de diretórios.

O Linux é um sistema operacional multi-usuário. Todo o aspecto do sistema é multiusuário, inclusive o sistema de arquivos. O sistema armazena informações sobre quem possui um arquivo e quem pode ler este arquivo. Existem outras partes únicas sobre o sistema de arquivos, como os links e a montagem de partições NFS. Essa seção explica tudo isso, tão bem quanto os aspectos multi-usuário do sistema de arquivos.

9.1 Propriedade

O sistema de arquivos armazena informações sobre o dono de cada arquivo e diretório no sistema. Isso inclui qual usuário e grupo possui um determinado arquivo. A maneira mais fácil de ver essa informação é com o comando *ls*:

```
$ ls -l /usr/bin/wc
-rwxr-xr-x   1 root     bin    7368 Jul 30  1999 /usr/bin/wc
```

Nós estamos interessados na terceira e quarta colunas (da esquerda para a direita). Essas colunas contém o nome do usuário e grupo que possui o arquivo. Nós podemos ver que o usuário "*root*"e o grupo "*bin*"são os proprietários desse arquivo.

"Nós podemos mudar facilmente os donos de um arquivo com os comandos *chown*(1) (que significa mudar o proprietário") e *chgrp*(1) (que significa "mudar o grupo"). Para mudar o dono de um arquivo para *daemon*, nós podemos usar o comando *chown*:

```
# chown daemon /usr/bin/wc
```

Para mudar o grupo que possui um arquivo para "*root*", nós podemos usar o comando *chgrp*:

```
# chgrp root /usr/bin/wc
```

Nós podemos utilizar o comando *chown* para especificar as propriedades de usuário e grupo para um arquivo:

```
# chown daemon:root /usr/bin/wc
```

No exemplo acima, o usuário poderia ter usado um ponto, no lugar do dois-pontos. O resultado teria sido o mesmo; entretanto, o dois-pontos é considerado a melhor forma. O uso do ponto é obsoleto e poderá ser removido em futuras versões do comando *chown* para permitir nomes de usuários que contenham ponto. Esses nomes de usuário tendem a ser muito populares com Servidores Windows Exchange e são encontrados com maior freqüência em endereços de e-mail como: mr.jones@example.com. No Slackware, os administradores são alertados para evitarem esses nomes de usuários, porque alguns scripts utilizam o ponto para indicar o nome e o grupo de um arquivo ou diretório. Em nosso exemplo, o comando *chmod* interpretaria mr.jones como usuário "mr"e grupo "jones".

A definição de quem é o dono de um determinado arquivo é uma parte muito importante na utilização de um sistema Linux, mesmo que você seja o único usuário do sistema. Algumas vezes você precisa corrigir as propriedades em arquivos e em nós de dispositivos (device nodes).

9.2 Permissões

As permissões são outra parte importante do aspecto multi-usuário do sistema de arquivos. Com elas, você pode alterar quem lê, escreve, e executa arquivos.

A informação de permissão é armazenada na forma de quatro dígitos octais, cada um especificando diferentes conjuntos de permissões. São eles, permissão do proprietário, permissão de grupo e, permissão do mundo. Os quatro dígitos octais são utilizados para armazenar informações especiais como identificação de usuário (user ID), identificação de grupo (group ID), e o bit controle. Os valores octais associados aos modos de permissão são (eles possuem letras associadas que são mostradas por programas como o comando *ls* e podem ser utilizadas pelo comando *chmod*):

Valores de Permissão Octais

Tipo de Permissão	Valor Octal	Valor Letra
"bit"controle	1	t
identificação de usuário	4	s
identificação de grupo	2	s
leitura	4	r
escrita	2	w
execução	1	x

Você adiciona os valores octais para cada grupo de permissões. Por exemplo, se você quer que o grupo tenha permissão de "leitura"e "escrita", você utilizará "6"na porção referente à permissão do grupo.

As permissões padrões do *bash* são:

```
$ ls -l /bin/bash
-rwxr-xr-x   1 root     bin  477692 Mar 21 19:57 /bin/bash
```

O primeiro traço pode ser substituído por um "d"se for um diretório. As três permissões de grupo (proprietário, grupo, e outros) são mostradas a seguir. Nós vemos que o proprietário possui permissões de leitura, escrita e execução (rwx). O grupo possui apenas permissão de leitura e execução (r-x). E todo o mundo possui permissões de leitura e escrita (r-x).

Como nós configuramos as permissões de um outro arquivo para que elas fiquem semelhantes as permissões do *bash*? Primeiro, vamos criar um arquivo chamado exemplo:

```
$ touch /tmp/exemplo
$ ls -l /tmp/exemplo
-rw-rw-r---  1 david    users    0 Apr 19 11:21 /tmp/exemplo
```

Nós usaremos o comando *chmod*(1) (que significa "mudar o modo") para alterar as permissões do arquivo exemplo. Acrescente os números octais para a permissão que você quer. Para o proprietário que possui permissão de leitura, escrita e execução, nós teríamos o valor 7. Para permissão de leitura e execução teríamos o valor 5. Aplique esses valores juntos ao *chmod* dessa maneira:

```
$ chmod 755 /tmp/exemplo
$ ls -l /tmp/exemplo
-rwxr-xr-x   1 david    users    0 Apr 19 11:21 /tmp/exemplo
```

Agora você deve estar pensando, "Por que nós não criamos o arquivo com essas permissões desde o início?"Bem, a resposta é simples. O *bash* possui uma implementação chamada *umask*. Este comando vem incluído com a maioria dos shells do Unix, e controla quais permissões são designadas à arquivos criados recentemente. Nós discutimos sobre as implementações do *bash* com maior profundidade na seção 8.3.1. O comando *umask* é utilizado de forma razoável. Ele funciona de maneira bastante similar ao *chmod*, só que em modo reverso. Você especifica os valores octais que você não deseja que estejam presentes nos arquivos recém-criados. O seu valor padrão é 0022.

```
$ umask
0022
$ umask 0077
$ touch tempfile
$ ls -l tempfile
-rw--------  1 david    users    0 Apr 19 11:21 tempfile
```

Olhe a página do manual do *bash* para maiores informações.

Para aplicar permissões especiais com o *chmod*, adicione os números juntos e identifique-os na primeira coluna. Por exemplo, para forçar a identificação do usuário (user ID) e a identificação de grupo (group ID), nós usamos o 6 na primeira coluna:

```
$ chmod 6755 /tmp/exemplo
$ ls -l /tmp/exemplo
-rwsr-sr-x   1 david    users    0 Apr 19 11:21 /tmp/exemplo
```

Se os valores octais confundem você, você pode usar letras com o *chmod*. Os grupos de permissão são representados como:

Proprietário	u
Grupo	g
Mundo	o
Todas acima	a

Para fazer o descrito acima, nós precisamos utilizar várias linhas de comando:

```
$ chmod a+rx /tmp/exemplo
$ chmod u+w /tmp/exemplo
$ chmod ug+s /tmp/exemplo
```

Algumas pessoas preferem utilizar as letras em vez dos números. Ambas as maneiras irão gerar o mesmo resultado na aplicação das permissões.

O formato octal é freqüentemente mais rápido, e o que você vê frequentemente sendo utilizado em shell script. Em alguns casos as letras são mais poderosas. Por exemplo, não existe uma maneira fácil de alterar um grupo de permissões e preservar outros em arquivos e diretórios, quando você usa o formato octal. Isso é trivial utilizando as letras.

```
$ ls -l /tmp/
-rwxr-xr-x   1 alan    users    0 Apr 19 11:21 /tmp/exemplo0
-rwxr-x---   1 alan    users    0 Apr 19 11:21 /tmp/exemplo1
----r-xr-x   1 alan    users    0 Apr 19 11:21 /tmp/exemplo2

$ chmod g-rwx /tmp/exemplo
-rwx---r-x   1 alan    users    0 Apr 19 11:21 /tmp/exemplo0
-rwx------   1 alan    users    0 Apr 19 11:21 /tmp/exemplo1
-------r-x   1 alan    users    0 Apr 19 11:21 /tmp/exemplo2
```

Nós mencionamos as permissões de identificação de usuário (user ID) e a identificação de grupo (group ID) em várias partes acima. Você pode estar se perguntando o que são essas permissões. Normalmente, quando você roda um programa, este está operando sob a sua conta de usuário. Isto é, ele tem todas as permissões que o seu usuário possui. O mesmo é verdade para o grupo. Quando você roda um programa, ele é executado sob o seu grupo atual. Com a permissão de identificação de usuário, você pode forçar o programa para sempre rodar como o proprietário do programa (como o ”root”). A permissão de identificação de grupo faz a mesma coisa, só que para o grupo.

Seja cuidadoso com isso, pois as permissões de identificações de usuário e de grupo dos programas podem abrir brechas de segurança no seu sistema. Se você freqüentemente força as permissões de usuário para os programas que pertencem ao usuário *root*, você está permitindo que qualquer um utilize o programa e o utilize como *root*. Como o usuário *root* não possui restrições no sistema, você pode ver como isso cria um grande problema de segurança. Resumindo, não é ruim utilizar as permissões de identificação de usuário e de grupo, apenas use o bom senso.

9.3 Links

Links são ponteiros entre arquivos. Com os links, você pode ter arquivos existindo em vários locais e podendo ser acessados por vários nomes. Existem dois tipos de links: estáticos e dinâmicos.

Links estáticos são nomes para um arquivo em particular. Eles só podem existir dentro de um único sistema de arquivos e só são removidos quando o nome real é removido do sistema. Eles são úteis em alguns casos, porém muitos usuários acham o link dinâmico muito mais versátil.

O link dinâmico, também chamado de link simbólico, pode apontar para um arquivo fora do seu sistema de arquivos. Ele é na verdade um pequeno arquivo contendo apenas as informações que ele necessita. Você pode adicionar ou remover links simbólicos sem afetar o arquivo atual. E desde que um link simbólico é na verdade um arquivo pequeno contendo sua própria informação, ele também pode apontar para um diretório. É bastante comum ver o */var/tmp* atualmente como um link simbólico para */tmp* por exemplo.

Os links não possuem seu próprio conjunto de permissões ou propriedades, porém refletem os arquivos aos quais eles apontam. Os usuários do Slackware usam mais os links dinâmicos. Aqui está um exemplo comum:

```
$ ls -l /bin/sh
lrwxrwxrwx   1 root     root      4 Apr  6 12:34 /bin/sh -> bash
```

O shell *sh* sob o Slackware é atualmente o *bash*. A remoção de links é feita utilizando o comando *rm*. O comando *ln* é utilizado para criar links. Esses comandos serão discutidos com maior profundidade no Capítulo 10.

É muito importante ter cuidado, em particular, com os links simbólicos. Uma vez, eu estava trabalhando em uma máquina que teimava em falhar nos backups em fita toda noite. Dois links simbólicos tinham sido feitos para dois diretórios, um embaixo do outro. O software para backup manteve anexado à fita esses mesmos diretórios, até ficar sem espaço. Normalmente, uma série de verificações prévias irá prever a criação de links simbólicos nessa situação, mas o nosso era um caso especial.

9.4 Montando Dispositivos

Como foi préviamente discutido na seção 4.1.1, todos os drives e dispositivos em seu computador são um sistema de arquivos enorme. Várias partições em seu disco rígido, CD-ROMs, e disquetes são todos alocados em uma mesma árvore. Ao anexar todos esses drives ao sistema de arquivos para que possam ser acessados, você precisa utilizar os comandos *mount*(1) e *umount*(1).

Alguns dispositivos são montados automaticamente quando você inicializa o seu computador. Esses dispositivos estão listados no arquivo */etc/fstab*. Tudo o que você quiser que seja montado automaticamente precisa ter uma entrada neste arquivo. Para os outros dispositivos, você terá que executar um comando toda vez que você quiser montar um dispositivo.

fstab

Vamos dar uma olhada em um exemplo do arquivo */etc/fstab*:

```
$ cat /etc/fstab

/dev/sda1 /               ext2      defaults        1   1
/dev/sda2 /usr/local      ext2      defaults        1   1
/dev/sda4 /home           ext2      defaults        1   1
/dev/sdb1 swap            swap      defaults        0   0
/dev/sdb3 /export         ext2      defaults        1   1
none      /dev/pts        devpts    gid=5,mode=620  0   0
```

```
none        /proc          proc        defaults        0   0
/dev/fd0    /mnt           ext2        defaults        0   0
/dev/cdrom/mnt/cdrom       iso9660     ro              0   0
```

A primeira coluna é o nome do dispositivo. Nesse caso, os dispositivos são cinco partições espalhadas por dois discos rígidos SCSI, dois sistemas de arquivos que não precisam de dispositivos, um drive de disquete, e um drive de CD-ROM. A segunda coluna informa aonde o dispositivo será montado. Esse deve ser o nome de um diretório, com exceção no caso de uma partição swap. A terceira coluna é o tipo de sistema de arquivo do dispositivo. Para os sistemas de arquivos normais do Linux, estes serão ext2 (segundo sistema de arquivos extendido). Drives de CD-ROM são iso9660, e dispositivos baseados no Windows serão do tipo msdos ou vfat.

A quarta coluna é uma listagem de opções que se aplicam ao sistema de arquivos montado. O padrão (defaults) pode ser usado sem problemas com quase todos os dispositivos. Entretanto, para dispositivos de somente leitura você deve colocar a opção ro. Existem muitas opções que podem ser utilizadas. Verifique o manual do *fstab*(5) para maiores informações. As duas últimas colunas são utilizadas para o *fsck* e outros comandos que precisam manipular os dispositivos. Verifique o manual para essas informações.

Quando você instala o Slackware Linux, o programa de instalação irá criar a maior parte do arquivo *fstab*.

mount e umount

Anexar outro dispositivo ao seu sistema de arquivos é fácil. Tudo o que você tem que fazer é utilizar o comando *mount*, com algumas opções. Utilizar o comando *mount* pode ser mais simples se o dispositivo possui uma entrada no arquivo */etc/fstab*. Por exemplo, vamos dizer que eu queira montar o meu driver de CD-ROM e o arquivo *fstab* se parece com o exemplo citado na seção anterior. Eu poderia utilizar o comando *mount* desta forma:

```
$ mount /cdrom
```

Desde que haja uma entrada no arquivo *fstab* para esse ponto de montagem, o comando *mount* sabe quais opções utilizar. Caso não haja uma entrada para este dispositivo, eu terei de utilizar muitas opções junto ao comando *mount*:

```
$ mount -t iso9660 -o ro /dev/cdrom /cdrom
```

Esta linha de comando inclui a mesma informação que o exemplo do *fstab* tinha, mesmo assim nós iremos explicar todas as partes do comando. A opção <option>-t iso9660</option> é o tipo de sistema de arquivos do dispositivo a ser montado. Neste caso, ele seria o sistema de arquivos iso9660, que é usado normalmente pelos drivers de CD-ROM. A opção <option>-o ro</option> diz ao comando para montar o dispositivo com capacidade para apenas efetuar leituras. O caminho */dev/cdrom* é o nome do dispositivo a ser montado, e o diretório */cdrom* é o local no sistema de arquivos onde o drive será montado.

Antes que você possa remover um disquete, CD-ROM, ou outro dispositivo removível, que esteja atualmente montado, você terá que desmontá-lo. Isto é feito utilizando o comando *umount*. Não pergunte para onde o ”n”foi porque nós não poderíamos te contar. Você pode utilizar, tanto o caminho para o dispositivo montado quanto o ponto de montagem, como argumento para o

comando *umount*. Por exemplo, se você quer desmontar o CD-ROM do exemplo anterior, ambos os comandos podem ser utilizados:

```
# umount /dev/cdrom
# umount /cdrom
```

9.5 Montagem NFS

NFS significa Network Filesystem (Sistema de arquivos de rede). Ele não é realmente parte do sistema de arquivos real, mas pode ser utilizado para adicionar partes ao sistema de arquivos montado.

Ambientes Unix enormes, algumas vezes, precisam compartilhar os mesmos programas, conjuntos de diretórios de usuários, e o repositório de e-mail. O problema de pegar a mesma cópia para cada máquina é resolvido pelo NFS. Nós podemos utilizar o NFS para compartilhar um conjunto de diretórios de usuários entre todas as estações de trabalho. Então as estações de trabalho montam o diretório que o NFS compartilhou como se estivesse nas suas próprias máquinas.

Veja a seção 5.6.2 e as páginas do manual para *exports*(5), *nfsd*(8), e *mountd*(8) para maiores informações.

Capítulo 10

Manipulando arquivos e diretórios

O Linux tenta ser o mais parecido com o Unix possível.Tradicionalmente, os sistemas operacionais Unix tem sido orientados pela linha de comando. É claro que temos uma interface gráfica no Slackware, mas a linha de comando é ainda o principal nível de controle do sistema. Portanto, é importante entender alguns dos comandos básicos dos comandos de gerenciamento de arquivos.

A seção a seguir explica os comandos corriqueiros do gerenciamento de arquivos e provê exemplos de como eles são usados. Existem muitos outros comandos, mas estes irão ajudar você a começar. Além disso, os comandos estão explicados resumidamente. Você encontrará mais detalhes nas man pages de cada comando.

10.1 Navegação: ls, cd, e pwd

10.1.1 ls

Esse comando lista arquivos em um diretório. Usuários DOS e Windows irão notar a similaridade com o comando dir. Ele mesmo, ls(1) irá listar os arquivos no diretório atual. Para ver o que existe em seu diretório raiz, você poderia usar esses comandos:

```
$ cd /
$ ls
bin   cdr    dev  home  lost+found  proc  sbin   tmp  var
boot  cdrom  etc  lib   mnt   root  suncd  usr  vmlinuz
```

O problema que muita gente tem com a saída, do comando anterior, é que não se pode dizer facilmente o que é um diretório e o que é um arquivo. Alguns usuários preferem o ls adicionado a um identificador para cada listagem, como essa:

```
$ ls -FC
bin/   cdr/    dev/  home/  lost+found/  proc/  sbin/   tmp/  var/
boot/  cdrom/  etc/  lib/   mnt/   root/  suncd/  usr/  vmlinuz
```

Diretórios têm uma barra (/) no final do nome, arquivos executáveis têm um asterisco no final do nome, e assim por diante.

O *ls* pode também ser usado para conseguir outras informações nos arquivos. Por exemplo, para ver a data de criação, proprietário, e permissões, você especificaria uma lista longa:

```
$ ls -l
```

```
drwxr-xr-x   2 root     bin     4096 May  7 09:11 bin/
drwxr-xr-x   2 root     root    4096 Feb 24 03:55 boot/
drwxr-xr-x   2 root     root    4096 Feb 18 01:10 cdr/
drwxr-xr-x  14 root     root    6144 Oct 23 18:37 cdrom/
drwxr-xr-x   4 root     root   28672 Mar  5 18:01 dev/
drwxr-xr-x  10 root     root    4096 Mar  8 03:32 etc/
drwxr-xr-x   8 root     root    4096 Mar  8 03:31 home/
drwxr-xr-x   3 root     root    4096 Jan 23 21:29 lib/
drwxr-xr-x   2 root     root   16384 Nov  1 08:53 lost+found/
drwxr-xr-x   2 root     root    4096 Oct  6 12:47 mnt/
dr-xr-xr-x  62 root     root       0 Mar  4 15:32 proc/
drwxr-x--x  12 root     root    4096 Feb 26 02:06 root/
drwxr-xr-x   2 root     bin     4096 Feb 17 02:02 sbin/
drwxr-xr-x   5 root     root    2048 Oct 25 10:51 suncd/
drwxrwxrwt   4 root     root  487424 Mar  7 20:42 tmp/
drwxr-xr-x  21 root     root    4096 Aug 24 03:04 usr/
drwxr-xr-x  18 root     root    4096 Mar  8 03:32 var/
```

Suponhamos que você queira uma lista dos arquivos ocultos no diretório atual.O comando seguinte listaria isso.

```
$ ls -a
.         bin    cdrom  home        mnt    sbin    usr
..        boot   dev    lib         proc   suncd   var
.pwrchute_tmp  cdr    etc    lost+found  root   tmp     vmlinuz
```

Arquivos començando com um ponto (chamados de arquivos ponto) são ocultados quando você executa o *ls*. Você somente verá esses arquivos se passar a opção *-a*.

Existem muitas outras opções que podem ser encontradas nos manuais online. Não esqueça que pode combinar as opções que você passa para o *ls*.

10.1.2 cd

O comando *cd* é usado para mudar o diretório de trabalho. Você simplesmente digita *cd* seguido do nome do caminho para mudar. Aqui estão alguns exemplos:

```
darkstar:~$ cd /bin
darkstar:/bin$ cd usr
bash: cd: usr: No such file or directory
darkstar:/bin$ cd /usr
darkstar:/usr$ ls
bin
darkstar:/usr$ cd bin
darkstar:/usr/bin$
```

Note que sem começar com a barra (/) ele tenta mudar para um diretório no diretório atual. Também executando o *cd* sem nenhuma opção moverá você para o seu diretório pessoal.

O comando *cd* não é igual a outros comandos. Ele é um comando integrado do shell. Comandos integrados do shell serão discutidos na seção 8.3.1. Isso pode não fazer nenhum sentido pra você agora. Basicamente isso significa que não existe man page para esse comando. Ao

invés disso, você tem que usar o help do shell. Como segue:

```
$ help cd
```

Serão mostradas as opções para o cd e como usá-las.

10.1.3 pwd

O comando *pwd* é usado para mostrar a sua localização atual. Para usar o comando *pwd* basta digitar pwd. Por exemplo:

```
$ cd /bin
$ pwd
/bin
$ cd /usr
$ cd bin
$ pwd
/usr/bin
```

10.2 Paginadores: more, less e most

10.2.1 more

more (1) é como chamamos um utilitário de paginação. Muitas vezes a saída de um comando em particular é muito grande para caber em apenas uma tela. Os comandos individualmente não sabem como fazer caber as saídas em telas separadas. Eles deixam esse trabalho para os paginadores. O comando more quebra a saída numa tela única e espera que você pressione a barra de espaços antes de continuar na próxima página. Pressionando a tecla enter irá avançar a saída em uma linha. Aqui segue um bom exemplo:

```
$ cd /usr/bin
$ ls -l
```

Isso deveria rolar a tela um pouco. Para parar a saída tela por tela, pipe (|) seguido do more:

```
$ ls -l $|$ more
```

Esse é o caractere pipe (shift + \). O pipe é: pegue a saída do *ls* e alimente o *more*. Você pode "pipar"qualquer coisa através do comando *more*, não apenas o *ls*. Pipe é também visto na seção 8.2.3.

10.2.2 less

O comando *more* é mais simples, mas não permite que você volte para uma página já exibida. O *more* não provê um jeito de você voltar. O comando *less* (1) prove essa funcionalidade. Ele é usado do mesmo jeito que o comando *more*, então o exemplo anterior serve aqui também. Então, o *less* é mais que o *more*. Just Krammer define esses comandos com o seguinte trocadilho:

> "**less** is more, but more **more** than **more** is, so **more** is less **less**, so use more **less** if you want less **more**."

10.2.3 most

Onde *more* e o *less* não conseguem ir, o *most* (1) vai além. Se *less* é mais do que *more*, *most* é mais do que *less.* Enquanto os outros paginadores podem apenas mostrar um arquivo por vez, *most* é capaz de vizualizar qualquer número de arquivos, até que cada janela de arquivo tenha pelo menos duas linhas aparecendo. O comando *most* tem muitas opções, confira as páginas de manual para mais detalhes.

10.3 Saída Simples: cat e echo

10.3.1 cat

cat (1) é a abreviação de concatenar. Ele foi originalmente desenhado para mesclar arquivos de textos em outro, mas pode ser usado em muitos outros propósitos. Para mesclar dois ou mais arquivos em um, você simplesmente lista os arquivos depois do comando *cat* e então redireciona a saída para um novo arquivo. O comando *cat* funciona com entrada padrão e saída padrão, então você deve usar os caracteres de redirecionamento do shell. Por exemplo:

```
$ cat file1 file2 file3 > bigfile
```

Esse comando pega o conteúdo dos arquivos *file1*, *file2*, e *file3* e mescla. A nova saída é enviada para a saída padrão.

O comando *cat* também pode ser usado para exibir arquivos. Muitas pessoas usam o *cat* em arquivos de textos junto com os comandos *more* ou *less*, como segue:

```
$ cat file1 $|$ more
```

Esse comando irá mostrar o arquivo *file1* e "pipá-lo"através do comando *more* para que você apenas veja uma tela por vez. Outro uso comum para o *cat* é copiar arquivos. Você pode copiar qualquer arquivo usando o *cat*, como segue:

```
$ cat /bin/bash $>$ \~{}/mybash
```

O programa */bin/bash* é copiado para o seu diretório */home* e nomeado como *mybash.*

O comando *cat* tem muitos usos e os mostrados aqui são apenas alguns. Desde que o *cat* faz uso extensivo de entradas e saídas padrão, ele é ideal para usar em scripts shell ou parte de outros comandos complexos.

10.3.2 echo

O comando *echo* (1) mostra o texto especificado na tela. Você precisa especificar a string que vai ser mostrada depois do comando. Por padrão o comando *echo* irá mostrar a string e imprimir a nova linha de caracteres depois dele. Você pode passar a opção *-n* para suprimir a impressão de uma nova linha. A opção *-e* vai causar a procura por caracteres escape na string e executá-las.

10.4 Criação: touch e mkdir

10.4.1 touch

O comando *touch* é usado para mudar a data de um arquivo. Você pode mudar a data de acesso e a data de modificação com esse comando. Se o arquivo especificado não existir, o comando *touch* criará um arquivo vazio. Para marcar um arquivo com a data atual, você poderia usar esse comando:

```
$ ls -al file1
-rw-r--r--    1 root     root  9779 Feb  7 21:41 file1
$ touch file1
$ ls -al file1
-rw-r--r--    1 root     root  9779 Feb  8 09:17 file1
```

Existem muitas outras opções para o comando *touch* para especificar qual data que você quer modificar, qual hora usar, e muitas outras opções.A man-page mostra isso em detalhes.

10.4.2 mkdir

O comando *mkdir* irá criar um novo diretório. Você simplesmente especifica o diretório a ser criado quando você executa o *mkdir*. Esse exemplo criará o diretório *hejaz* no diretório atual.

```
$ mkdir hejaz
```

Você também pode especificar o caminho, como segue:

```
$ mkdir /usr/local/hejaz
```

A opção *-p* irá dizer ao *mkdir* para criar qualquer diretório pai. O exemplo acima falhará se */usr/local* não existir. A opção *-p* irá criar */usr/local* e */usr/local/hejaz*:

```
$ mkdir -p /usr/local/hejaz
```

10.5 Copiar e mover

10.5.1 cp

O comando *cp* (1) copia arquivos. Usuários do DOS irão notar a similaridade com o comando *copy*. Existem muitas opções para o comando *cp*, então você deve dar uma olhada na páginal de manual antes de usá-lo.

Um uso comum é usar o *cp* para copiar um arquivo de uma localização para outra. Por exemplo:

```
$ cp hejaz /tmp
```

Isso copia o arquivo *hejaz* do diretório atual para o diretório */tmp*.

Muitos usuários preferem manter a data do arquivo, para isso:

```
$ cp -a hejaz /tmp
```

Isso assegura que a data não será modificada na cópia.

Para cópia recursiva do conteúdo de um diretório para outro, você usaria esse comando:

```
$ cp -R \textit{mydir} /tmp
```

Isso copiará o diretório *mydir* para o diretório */tmp*.

Também se você quiser copiar um diretório ou um arquivo e mante-lo com as permissões antigas, datas e deixa-lo exatamente o mesmo, use o comando *cp -p*.

```
$ ls -l file
-rw-r--r--    1 root     vlad      4 Jan  1 15:27 file
$ cp -p file /tmp
$ ls -l /tmp/file
-rw-r--r--    1 root     vlad      4 Jan  1 15:27 file
```

O comando *cp* tem muitas outras opções que são mostradas em detalhes nas páginas de manual.

10.5.2 mv

O comando *mv* (1) move os arquivos de um lugar para outro. Simples o bastante, não?

```
$ mv oldfile /tmp/newfile
```

O comando *mv* tem poucas opções na linha de comando que são detalhadas nas páginas de manual Na prática, o comando *mv* é raramente usado com opções de linha de comando.

10.6 Exclusão: rm e rmdir

10.6.1 rm

O comando *rm* (1) remove arquivos e árvores de diretório.

Os usuários do DOS notarão em ambos a similaridade com os comandos *del* e o *deltree*.

O comando *rm* pode ser muito perigoso se você não se cuidar. Enquanto as vezes é possível recuperar um arquivo removido, isso pode ser uma tarefa complicada (e certamente cara) além de estar fora do escopo desse livro.

Para remover um único arquivo, especifique o nome dele quando você executar o rm:

```
$ rm file1
```

Se o arquivo tiver as permissões de gravação, você poderá ter uma mensagem de erro de permissão negada. Para forçar a remoção do arquivo não importando com nada, passe a opção *-f*, como segue:

```
$ rm -f file1
```

Para remover um diretório inteiro, você pode usar o *-r* e o *-f* juntos. Esse é um bom exemplo de como remover conteúdos inteiros do seu hd. Você realmente não quer fazer isso. Mas existe um comando mesmo assim:

```
# rm -rf /
```

Seja cauteloso com o comando *rm*; você pode atirar no seu próprio pé. Existem várias opções na linha de comando, que são discutidas nas páginas de manual.

10.6.2 rmdir

O comando *rmdir* (1) remove diretórios do sistema de arquivos. O diretório deve estar vazio antes de poder ser removido. A síntaxe é simples:

```
$ rmdir $<$directory$>$
```

Esse exemplo irá remover o subdiretório *hejaz* no diretório atual:

```
$ rmdir hejaz
```

Se esse diretório não existir, o *rmdir* dirá a você. Você também pode especificar um caminho completo de um diretório a ser removido, como esse exemplo mostra:

```
$ rmdir /tmp/hejaz
```

Esse exemplo irá tentar remover o diretório *hejaz* dentro do diretório */tmp*.

Você também pode remover um diretório e todos os diretórios pai passando a opção *-p*.

```
$ rmdir -p /tmp/hejaz
```

Isso primeiro irá tentar remover o diretório *hejaz* dentro do */tmp*. Caso tenha sucesso, ele irá tentar remover o */tmp*. O comando *rmdir* irá continuar isso até ocorrer um erro ou até que a árvore especificada seja removida.

10.7 Apelidando arquivos com ln

O comando *ln* (1) é usado para criar links entre arquivos. Esses links podem ser hard links ou links simbólicos. A diferença entre os dois tipos de links serão discutidos na seção 9.3.
Se você quiser fazer um link simbólico do diretório */var/media/mp3* e colocar esse link no seu diretório *home*, você faria isso:

```
$ ln -s /var/media/mp3 \~{}/mp3
```

A opção *-s* diz para o *ln* fazer um link simbólico. A próxima opção é o caminho do link, e a opção final é como chamar o link. Nesse caso, o comando apenas irá criar um arquivo chamado *mp3* no seu diretório *home* que aponta para o */var/media/mp3*. Você pode chamar o link do que você quiser apenas mudando a última opção.

Fazer um hard link é tão fácil quanto. Tudo que você tem que fazer é tirar a opção *-s*. Hard links normalmente não podem se referir a diretórios ou sistema de arquivos. Para criar um hard link de */usr/bin/email* para */usr/bin/mutt*, apenas digite o trecho a seguir:

```
# ln /usr/bin/mutt /usr/bin/email
```

Capítulo 11

Controle de Processos

Cada programa que está sendo executado é chamado de processo. Estes processos vão desde o ambiente gráfico X até programas de sistema (chamados "daemons") que são iniciados durante o boot do computador. Cada processo pertence a um usuário em particular. Processos que são iniciados no boot geralmente são executados como usuário *root* ou *nobody*. Processos que você inicia pertencerão ao seu usuário. Processo iniciados por outros usuários pertencerão a eles.

Você tem controle sobre todos os processos iniciados por você. Adicionalmente, o usuário *root* tem controle sobre todos os processo do sistema, incluindo aqueles iniciados por outros usuários. Processos podem ser controlados e monitorados através de vários programas, bem como de comandos do shell.

11.1 Segundo Plano (background)

Os programas iniciados à partir da linha de comando são executados em primeiro plano (foreground). Isto permite ver toda a saída do programa e interagir com ele. Entretanto há diversas situações em que será necessário executar o programa sem que o terminal fique preso à sua execução. Chamamos isso de executar o programa em segundo plano, e há várias maneiras de fazê-lo.

A primeira maneira de se colocar um processo em segundo plano é adicionando o ê comercial (&) à linha de comando quando você iniciar o programa. Por exemplo, digamos que você queira usar o *amp*, um mp3 player da linha de comando, para tocar um diretório cheio de arquivos mp3, mas você precisará utilizar o mesmo terminal para executar outros comandos. A seguinte linha de comando irá iniciar o programa amp em segundo plano:

```
$ amp *.mp3 &
```

O programa será executado normalmente e você retornará ao prompt de comando.

Outra maneira de colocar um processo em segundo plano é fazê-lo durante a sua execução. Primeiro, inicie um programa. Durante sua execução, pressione **Control+z**. Isso suspenderá o processo. Suspender um processo significa pausar a sua execução. Ele momentaneamente pára de rodar, mas pode ser iniciado novamente a qualquer momento. Uma vez suspenso o processo, você retorna ao prompt de comando. Você inicia o processo suspenso em segundo plano digitando:

```
$ bg
```

Agora o processo suspenso está rodando em segundo plano.

11.2 Primeiro plano (foregrounding)

Se você necessitar interagir com um processo em segundo plano, é possível trazê-lo para primeiro plano. Se existe apenas um processo em segundo plano, você pode trazê-lo de volta digitando:

```
$ fg
```

Se o programa ainda não terminou sua execução, então ele tomará o terminal e você não terá mais acesso ao prompt de comando. Algumas vezes o programa terminará sua execução enquanto estiver em segundo plano. Neste caso, você receberá uma mensagem como esta:

```
    [1]+  Done          /bin/ls $LS_OPTIONS
```

Informando que o processo em segundo plano (neste caso o *ls* - que não é um exemplo muito interessante) foi terminado.

É possível ter vários processos em segundo plano de uma só vez. Quando isto acontece, será necessário saber qual processo você deseja trazer para o primeiro plano. Apenas digitando *fg* trará para o primeiro plano o último processo que foi colocado em segundo plano. E se você tiver uma lista enorme de processos em segundo plano? Por sorte, o bash possui um comando para listar esses processos. Ele é o *jobs* e o seu resultado é:

```
$ jobs
[1]   Stopped            vim
[2]-  Stopped            amp
[3]+  Stopped            man ps
```

Isto mostra uma lista de todos os processos que estão em segundo plano. Como você pode ver, todos estão parados. Isto significa que os processos estão suspensos. O número é um tipo de ID para todos os processos em segundo plano. O ID seguido de um sinal de adição (*man ps*) é o processo que será trazido para primeiro plano se você apenas digitar *fg*.

Se quiser trazer para primeiro plano o *vim*, você deve digitar:

```
$ fg 1
```

e o *vim* rapidamente retornará à console. Enviar processos para o segundo plano pode ser muito útil se você tiver apenas um terminal aberto em uma conexão discada. Será possível executar diversos programas naquele único terminal, periodicamente alternando entre eles.

11.3 ps

Agora você sabe alternar entre vários processos iniciados à partir da linha de comando. Sabe também que existem muitos processos em execução todo o tempo. Mas então como listar todos esses programas? Bem, você pode utilizar o comando *ps*(1). Este comando tem uma porção de opções, mas iremos apresentar apenas algumas das mais importantes. Para uma listagem

completa, veja as páginas de manual do ps. Sobre as páginas de manual veja detalhes em Seções das páginas de Manual.

Apenas digitando *ps* você terá uma lista dos programas em execução no seu terminal. Isto inclui os processos em primeiro plano (incluindo aí qualquer shell que você esteja usando e, é claro, o próprio *ps*). Também são listados os processos que você está executando em segundo plano. Muitas vezes essa lista será bem curta:

Resultado padrão do *ps*

```
$ ps
 PID TTY    TIME CMD
 7923 ttyp0    00:00:00 bash
 8059 ttyp0    00:00:00 ps
```

Embora não sejam muitos processos, a informação é interessante. Você terá as mesmas colunas usando o ps sem parâmetros, não importa quantos processos estejam em execução. Então, o que significa tudo aquilo?

Bem, o *PID* é o *ID do processo.* Todos os processos em execução têm um identificador único que vai de 1 até 32767. Para cada processo é atribuído o próximo PID livre. Quando um processo é finalizado (ou interrompido, como você verá na próxima seção), ele libera o PID. Quando o PID máximo é alcançado, o próximo será o menor valor que estiver livre.

A coluna *TTY* indica em qual terminal o processo está sendo executado. Executando o *ps* sem parâmetros serão listados apenas os programas em execução no terminal corrente, então todos os processos terão a mesma informação na coluna TTY. Como você pode ver, ambos processos listados são executados em *ttyp0.* Isso indica que estão sendo executados remotamente ou a partir de alguma variedade de terminal X.

A coluna *TIME* indica quanto tempo de CPU o processo tem consumido. Isso é diferente do tempo gasto para execução do processo. Lembre-se que o Linux é um sistema operacional multi-tarefa. Existem muitos processos em execução todo o tempo, e cada um desses utiliza uma pequena porção do tempo do processador. Então, a coluna TIME deve mostrar muito menos tempo para cada processo do que ele atualmente necessita para ser executado. Se você notar mais do que vários minutos na coluna TIME, isto pode significar que algo está errado.

Finalmente, a coluna *CMD* mostra o que é o programa. Ele apenas lista o nome básico do programa, sem nenhuma opção de linha de comando ou informação similar. Para obter essa informação será necessário usar uma das muitas opções do *ps.* Iremos discuti-las brevemente.

Você pode conseguir uma listagem completa do que está em execução no seu sistema usando a combinação adequada de opções. Isto provavelmente resultará em uma longa lista de processos (cinqüenta e cinco no meu laptop enquanto escrevo esta sentença), então abreviarei a saída:

```
$ ps -ax
 PID TTYSTAT   TIME COMMAND
   1 ?  S      0:03 init [3]
   2 ?  SW     0:13 [kflushd]
   3 ?  SW     0:14 [kupdate]
   4 ?  SW     0:00 [kpiod]
   5 ?  SW     0:17 [kswapd]
```

```
  11 ?  S       0:00 /sbin/kerneld
  30 ?  SW      0:01 [cardmgr]
  50 ?  S       0:00 /sbin/rpc.portmap
  54 ?  S       0:00 /usr/sbin/syslogd
  57 ?  S       0:00 /usr/sbin/klogd -c 3
  59 ?  S       0:00 /usr/sbin/inetd
  61 ?  S       0:04 /usr/local/sbin/sshd
  63 ?  S       0:00 /usr/sbin/rpc.mountd
  65 ?  S       0:00 /usr/sbin/rpc.nfsd
  67 ?  S       0:00 /usr/sbin/crond -l10
  69 ?  S       0:00 /usr/sbin/atd -b 15 -l 1
  77 ?  S       0:00 /usr/sbin/apmd
  79 ?  S       0:01 gpm -m /dev/mouse -t ps2
  94 ?  S       0:00 /usr/sbin/automount /auto file /etc/auto.misc
 106 tty1      S0:08 -bash
 108 tty3      SW     0:00 [agetty]
 109 tty4      SW     0:00 [agetty]
 110 tty5      SW     0:00 [agetty]
 111 tty6      SW     0:00 [agetty]
 [output cut]
```

Muitos desses processos são iniciados durante o boot na maioria dos sistemas. Eu fiz algumas poucas modificações no meu sistema, então isso poderá variar dependendo da sua quilometragem. Entretanto, você verá muitos desses processos no seu sistema também. Como você pode ver, essas opções exibem as opções da linha de comando que originou o processo em execução. Recentemente, uma vulnerabilidade do kernel no *ptrace* fez com que não seja permitida a exibição das opções de linha de comando de muitos processos em execução. Estes são listados agora entre colchetes como dos PIDs 108 ao 110. Também temos algumas colunas a mais e outros dados interessantes.

Primeiro, você perceberá que muitos desses processos são listados como em execução no tty "?". Esses não estão anexados a nenhum terminal em particular. Isto é muito comum com processos executados pelo sistema ("daemons"), os quais não são vinculados a nenhum terminal. "Daemons"comuns são sendmail, BIND, apache e NFS. Eles, de um modo em geral, aguardam pela requisição de um cliente, retornando uma resposta assim que são demandados.

Segundo, há uma nova coluna: *STAT*. Ela mostra o status do processo. *S* corresponde a parado ("sleeping"): o processo está esperando algo acontecer. *Z* corresponde a um processo zumbi. Um processo zumbi é aquele cujo pai (isto é, o processo que o originou) morreu, deixando o processo filho para trás. Isso não é uma coisa boa. *D* corresponde a um processo que está irreversivelmente parado. Freqüentemente esses processos recusam-se a morrer, mesmo quando recebem um SIGKILL. Você pode ler mais a respeito do SIGKILL na próxima seção sobre o *kill*. O *W* corresponde a esperando uma chamada. Um processo morto é indicado com um *X*. Um processo indicado por um *T* é rastreado ou parado. *R* significa que o processo é executável.

Se você deseja ver ainda mais informações sobre os processos em execução, tente isto:

```
$ ps -aux
 USER PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
 root   1  0.0  0.0   344   80 ?        S    Mar02   0:03 init [3]
 root   2  0.0  0.0     0    0 ?        SW   Mar02   0:13 [kflushd]
 root   3  0.0  0.0     0    0 ?        SW   Mar02   0:14 [kupdate]
```

```
root   4  0.0  0.0     0    0 ?        SW   Mar02   0:00 [kpiod]
root   5  0.0  0.0     0    0 ?        SW   Mar02   0:17 [kswapd]
root  11  0.0  0.0  1044   44 ?        S    Mar02   0:00 /sbin/kerneld
root  30  0.0  0.0  1160    0 ?        SW   Mar02   0:01 [cardmgr]
bin   50  0.0  0.0  1076  120 ?        S    Mar02   0:00 /sbin/rpc.port
root  54  0.0  0.1  1360  192 ?        S    Mar02   0:00 /usr/sbin/sysl
root  57  0.0  0.1  1276  152 ?        S    Mar02   0:00 /usr/sbin/klog
root  59  0.0  0.0  1332   60 ?        S    Mar02   0:00 /usr/sbin/inet
root  61  0.0  0.2  1540  312 ?        S    Mar02   0:04 /usr/local/sbi
root  63  0.0  0.0  1796   72 ?        S    Mar02   0:00 /usr/sbin/rpc.
root  65  0.0  0.0  1812   68 ?        S    Mar02   0:00 /usr/sbin/rpc.
root  67  0.0  0.2  1172  260 ?        S    Mar02   0:00 /usr/sbin/cron
root  77  0.0  0.2  1048  316 ?        S    Mar02   0:00 /usr/sbin/apmd
root  79  0.0  0.1  1100  152 ?        S    Mar02   0:01 gpm
root  94  0.0  0.2  1396  280 ?        S    Mar02   0:00 /usr/sbin/auto
chris106  0.0  0.5  1820  680 tty1     S    Mar02   0:08 -bash
root 108  0.0  0.0  1048    0 tty3     SW   Mar02   0:00 [agetty]
root 109  0.0  0.0  1048    0 tty4     SW   Mar02   0:00 [agetty]
root 110  0.0  0.0  1048    0 tty5     SW   Mar02   0:00 [agetty]
root 111  0.0  0.0  1048    0 tty6     SW   Mar02   0:00 [agetty]
[output cut]
```

Isso é que é quantidade de informação! Basicamente foram incluídas informações como qual o usuário iniciou o processo, quanto o processo está usando dos recursos do sistema (as colunas %CPU, %MEM, VSZ e RSS), e em qual data o processo foi iniciado. Obviamente, isso é uma grande quantidade de informação que pode estar à mão para um administrador de sistemas. Isso também traz à tona outro ponto: a informação exibida é limitada à largura da tela o que não permite que você a veja completamente. A opção *-w* irá forçar o *ps* a quebrar linhas longas.

Não é a coisa mais linda, mas funciona. Agora você tem uma lista de informações completa para cada processo. Ainda há mais informações que podem ser mostradas para cada processo. Verifique a profunda página de manual do *ps*. Entretanto as opções mostradas acima são as mais populares e serão usadas com mais freqüência.

11.4 kill

Em algumas ocasiões, programas comportam-se mal e você precisará coloca-los ”na linha”. O programa para esse tipo de administração é chamado *kill*(1), e pode ser usado para manipular processos de várias maneiras. O mais óbvio dos usos do *kill* é interromper um processo. Você precisará fazer isso se um programa se perder e utilizar uma grande quantidade de recursos do sistema, ou se você só estiver cansado de vê-lo em execução.

Para interromper um processo você precisará saber seu PID ou seu nome. Para conseguir o PID utilize o comando *ps* que foi discutido na última seção. Por exemplo, para interromper o processo 4747, você deverá fazer o seguinte:

```
$ kill 4747
```

Perceba que você terá de ser o dono do processo para poder interrompê-lo. Isso é uma questão de segurança. Se você tivesse permissão para interromper processos iniciados por outros

usuários, seria possível realizar uma porção de coisas maliciosas. É claro, o *root* pode interromper qualquer processo do sistema.

Há ainda outra variação do comando *kill* chamada *killall*(1). Este programa faz exatamente o que diz: interrompe todos os processos que possuem determinado nome. Se você quisesse interromper todos os processos *vim* em execução, você poderia digitar o seguinte comando:

```
$ killall vim
```

Todo e qualquer processo *vim* que você estiver executando será interrompido. Fazendo isto como usuário *root* irá interromper todos os processos *vim* em execução, independente do usuário. Isso traz uma interessante maneira de derrubar todos usuários (incluindo você) do sistema:

```
# killall bash
```

Algumas vezes um simples kill não é suficiente. Certos processos não são interrompidos com um kill. Você terá de usar uma forma mais potente. Se aquele PID 4747 pentelho não responder ao seu kill, você poderá fazer o seguinte:

```
$ kill -9 4747
```

É quase certo que o processo 4747 será interrompido. Você pode fazer a mesma coisa com o *killall*. O que isso faz é enviar um sinal diferente para o processo. Um *kill* regular envia um sinal *SIGTERM* (terminar) para o processo, o qual diz que é para terminar o que estiver fazendo, fechar o que estiver utilizando e sair. *kill -9* envia um sinal *SIGKILL* (interromper) para o processo, o que essencialmente o para. Não é permitido ao processo fechar o que estiver utilizando e algumas vezes coisas ruins, como corrupção de dados, podem ocorrer com um *SIGKILL*. Há uma lista completa de sinais à sua disposição. Você pode obtê-la digitando:

```
$ kill -l
 1) SIGHUP      2) SIGINT     3) SIGQUIT    4) SIGILL
 5) SIGTRAP     6) SIGABRT    7) SIGBUS     8) SIGFPE
 9) SIGKILL    10) SIGUSR1   11) SIGSEGV   12) SIGUSR2
13) SIGPIPE    14) SIGALRM   15) SIGTERM   17) SIGCHLD
18) SIGCONT    19) SIGSTOP   20) SIGTSTP   21) SIGTTIN
22) SIGTTOU    23) SIGURG    24) SIGXCPU   25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF   28) SIGWINCH  29) SIGIO
30) SIGPWR
```

O número deve ser usado para o *kill*, enquanto o nome menos o prefixo "SIG"pode ser usado com o *killall*. Aqui vai outro exemplo:

```
$ killall -KILL vim
```

Um último uso para o *kill* é reiniciar um processo. Enviando um *SIGHUP* irá fazer com que muitos processos releiam seus arquivos de configuração. Isso é especialmente útil para fazer com que os processos releiam seus arquivos de configuração recém editados.

11.5 top

Finalmente, há um comando que você pode usar para exibir informações atualizadas sobre os processos em execução no sistema. Esse comando é chamado *top*(1), e é iniciado assim:

```
$ top
```

Isto irá exibir uma tela cheia com informações sobre os processos em execução no sistema, bem como algumas informações gerais sobre o próprio sistema. Isso inclui a carga média, número de processos, o status da CPU, informações sobre memória livre e detalhes dos processos incluindo PID, usuário, prioridade, uso de CPU e memória, tempo de execução e nome do programa.

```
 6:47pm  up 1 day,  18:01,  1 user,  load average: 0.02, 0.07, 0.02
61 processes: 59 sleeping, 2 running, 0 zombie, 0 stopped
CPU states:  2.8% user,  3.1% system,  0.0% nice, 93.9% idle
Mem:   257992K av, 249672K used,    8320K free,   51628K shrd,  78248K buff
Swap:   32764K av,     136K used,   32628K free,          82600K cached


 PID USER    PRI  NI  SIZE  RSS  SHARE  STAT  LIB %CPU %MEM  TIME COMMAND

 112 root     12   0 19376  18M   2468  R 0   3.7  7.5 55:53 X
4947 david    15   0  2136 2136   1748  S 0   2.3  0.8  0:00 screenshot
3398 david     7   0 20544  20M   3000  S 0   1.5  7.9  0:14 gimp
4946 root     12   0  1040 1040    836  R 0   1.5  0.4  0:00 top
 121 david     4   0   796  796    644  S 0   1.1  0.3 25:37 wmSMPmon
 115 david     3   0  2180 2180   1452  S 0   0.3  0.8  1:35 wmaker
4948 david    16   0   776  776    648  S 0   0.3  0.3  0:00 xwd
   1 root1   0   176  176    148  S       0   0.1  0.0  0:13 init
 189 david     1   0  6256 6156   4352  S 0   0.1  2.4  3:16 licq
4734 david     0   0  1164 1164    916  S 0   0.1  0.4  0:00 rxvt
   2 root0   0     0    0      0  SW      0   0.0  0.0  0:08 kflushd
   3 root0   0     0    0      0  SW      0   0.0  0.0  0:06 kupdate
   4 root0   0     0    0      0  SW      0   0.0  0.0  0:00 kpiod
   5 root0   0     0    0      0  SW      0   0.0  0.0  0:04 kswapd
  31 root0   0   340  340    248  S       0   0.0  0.1  0:00 kerneld
  51 root0   0    48   48     32  S       0   0.0  0.0  0:00 dhcpcd
  53 bin 0   0   316  316    236  S       0   0.0  0.1  0:00 rpc.portmap
  57 root0   0   588  588    488  S       0   0.0  0.2  0:01 syslogd
```

Ele é chamado de *top* porque os programas que fazem uso intensivo da CPU serão listados no topo. Uma observação interessante é que o próprio top será listado primeiro em muitos sistemas inativos (e alguns ativos) por causa da utilização de CPU. Entretanto, o *top* é muito útil para determinar qual programa está se comportando mal e precisa ser interrompido.

Mas suponha que você só queira uma lista dos seus próprios processos, ou dos processos de algum outro usuário. Os processos que você quer ver podem não estar entre os programas em execução com maior consumo de CPU. A opção *-u* permite que você especifique um nome de usuário ou UID e monitore somente os processos pertencentes àquele UID.

```
$ top -u alan
PID USERPR  NI  VIRT RES   SHR S %CPU %MEM   TIME+   COMMAND
3622 alan13   0 11012  10m 6956 S  1.0  2.1   0:03.66 gnome-terminal
3739 alan13   0  1012 1012  804 R  0.3  0.2   0:00.06 top
3518 alan 9   0  1312 1312 1032 S  0.0  0.3   0:00.09 bash
3529 alan 9   0   984  984  848 S  0.0  0.2   0:00.00 startx
3544 alan 9   0   640  640  568 S  0.0  0.1   0:00.00 xinit
3548 alan 9   0  8324 8320 6044 S  0.0  1.6   0:00.30 gnome-session
3551 alan 9   0  7084 7084 1968 S  0.0  1.4   0:00.50 gconfd-2
3553 alan 9   0  2232 2232  380 S  0.0  0.4   0:00.05 esd
3555 alan 9   0  2552 2552 1948 S  0.0  0.5   0:00.10 bonobo-activati
3557 alan 9   0  2740 2740 2224 S  0.0  0.5   0:00.05 gnome-smproxy
3559 alan 9   0  6496 6492 5004 S  0.0  1.3   0:00.31 gnome-settings-
3565 alan 9   0  1740 1740 1440 S  0.0  0.3   0:00.28 xscreensaver
3568 alan 9   0  7052 7052 4960 S  0.0  1.4   0:02.28 metacity
3572 alan 9   0 11412  11m 7992 S  0.0  2.2   0:01.58 gnome-panel
3574 alan 9   0 12148  11m 8780 S  0.0  2.4   0:00.64 nautilus
3575 alan 9   0 12148  11m 8780 S  0.0  2.4   0:00.00 nautilus
3576 alan 9   0 12148  11m 8780 S  0.0  2.4   0:00.00 nautilus
```

Como você pode ver, atualmente estou executando *X*, *top*, um *gnome-terminal* (no qual estou escrevendo isto) e muitos outros processos relacionados ao ambiente gráfico X, os quais estão tomando a maior parte do tempo da minha CPU. Isto é uma boa maneira de monitorar o quão duro seus usuários estão trabalhando no sistema.

O *top* também permite monitorar processos por seus PID, ignorando processos sem atividade e zumbis, e muitas outras opções. O melhor lugar para aprender a lidar com essas opções são as páginas de manual para o *top*.

Capítulo 12

O Essencial em Administração de Sistema

Ha Ha Ha Ha.... Eu sei o que você deve estar pensando."Eu não sou um administrador de sistema! Eu não quero mesmo ser um!"

O fato é que você é o administrador de qualquer computador no qual você possui a senha do usuário *root*. Pode ser seu desktop com um ou dois usuários, ou pode ser um grande servidor com várias centenas de usuários. Independente disto, você deverá saber como gerenciar usuários e como desligar o sistema de modo seguro. Estas tarefas parecem simples, mas elas têm alguns caprichos que você deve ter em mente.

12.1 Usuários e Grupos

Como mencionado no capítulo 8, você não deve usar normalmente seu sistema logado como *root*. Em vez disto, você deve criar uma conta de usuário normal para o uso diário e usar a conta do root apenas para tarefas de administração do sistema. Para criar um usuário, você pode usar as ferramentas fornecidas pelo Slackware ou editar manualmente o arquivo de senhas.

12.1.1 Scripts fornecidos

O modo mais fácil de gerenciar usuários e grupos é com os scripts e programas fornecidos. O Slackware inclui os programas *adduser*, *userdel*(8), *chfn*(1), *chsh*(1), e *passwd*(1) para gerenciar usuários. Os comandos *groupadd*(8), *groupdel*(8), e *groupmod*(8) são para gerenciar grupos. Com exceção de *chfn*, *chsh*, e *passwd*, esses programas são executados apenas pelo *root*, e estão, portanto, localizados em */usr/sbin*. *chfn*, *chsh*, e *passwd* podem ser executados por qualquer um e estão localizados em */usr/bin*.

Usuários podem ser adicionados com o comando *adduser*. Iniciaremos pelo procedimento completo, mostrando todas as perguntas que são solicitadas e uma breve descrição do que cada uma significa. A resposta padrão está entre colchetes e pode ser adotada para quase todas as perguntas, a menos que você realmente queira mudar alguma coisa.

```
# adduser
Login name for new user []: jellyd
```

Este é o nome que o usuário usará para se logar. Tradicionalmente, login names possuem oito ou menos caracteres e todos minúsculos (você pode usar mais que oito caracteres, inclusive numérico, mas evite fazer isso a menos que você tenha uma boa razão.)

Você pode fornecer o nome de login como um argumento do comando:

```
# adduser jellyd
```

No outro caso, depois de fornecer o nome de login, o comando adduser perguntará pelo ID do usuário:

```
User ID ('UID') [ defaults to next available ]:
```

O ID do usuário (UID) é quem realmente identifica os membros no Linux. Cada usuário tem um número único, começando em 1000 no Slackware. Você pode definir um UID para um novo usuário ou então deixar que o adduser designe para o usuário o próximo UID livre.

```
Initial group [users]:
```

Todos os usuários pertencem ao grupo *users* por padrão. Você pode querer colocar o usuário novo em um grupo diferente, mas isto não é recomendado a menos que você saiba o que está fazendo.

```
Additional groups (comma separated) []:
```

Esta pergunta permite você colocar o novo usuário em grupos adicionais. É possível para um usuário estar em vários grupos ao mesmo tempo. Isso é útil se você tem grupos específicos, por exemplo, modificar arquivos de web site, jogar, etc. Por exemplo, alguns site definem o grupo *wheel* como o único grupo que pode usar o comando *su*. Ou, como padrão, a instalação do Slackware usa o grupo *sys* para usuários autorizados a utilizarem a placa de som interna.

```
Home directory [/home/jellyd]
```

Os diretórios home padrão são colocados em */home*. Se você usa um sistema muito grande, é possível que você tenha que mover os diretórios home para uma localização diferente (ou para várias localizações). Esta etapa permite que você especifique onde será o diretório home do usuário.

```
Shell [ /bin/bash ]
```

bash é o shell padrão no Slackware Linux e será adequado para a maioria das pessoas. Se seu novo usuário vem de um ambiente Unix, ele pode estar familiarizado com um shell diferente. Você pode mudar o shell dele nesse momento ou ele pode mudar mais tarde usando o comando *chsh*.

```
Expiry date (YYYY-MM-DD) []:
```

As contas podem ser configuradas para expirarem numa data específica. Por padrão, não há uma data limite. Você pode mudar isso se você desejar. Essa opção pode ser útil para pessoas executando um ISP (Provedor de Serviços de Internet) que querem fazer uma conta expirar após uma certa data, a menos que elas recebam o pagamento do próximo ano.

```
New account will be created as follows:
---------------------------------------
Login name:   jellyd
UID:          [ Next available ]
Initial group:users
Additional groups:  [ None ]
Home directory:     /home/jellyd
Shell:        /bin/bash
Expiry date:  [ Never ]
```

Isto é ... se você quiser desistir, tecle **Control+C**. Caso contrário, tecle ENTER para seguir em frente e criar a conta.

Nesse momento, você vê todas as informações digitadas para a nova conta e tem a oportunidade de abortar a criação da conta. Se você entrou com algo incorreto, pode teclar **Control+C** e iniciar novamente. Caso contrário, você pode teclar enter e a conta será criada.

```
Creating new account...

Changing the user information for jellyd
Enter the new value, or press return for the default
  Full Name []: Jeremy
  Room Number []: Smith 130
  Work Phone []:
  Home Phone []:
  Other []:
```

Todas essas informações são opcionais. Você não tem que entrar com nenhuma dessas informações se não quiser, e o usuário pode mudar esses dados a qualquer hora usando o comando *chfn*. Entretanto, pode ser útil colocar pelo menos o nome completo e um número de telefone, no caso de você precisar entrar em contato com a pessoa em questão.

```
Changing password for jellyd
Enter the new password (minimum of 5, maximum of 127 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:
Password changed.

Account setup complete.
```

Você terá que entrar com uma senha para o novo usuário. Geralmente, se o usuário não estiver fisicamente presente nesse momento, você apenas colocará uma senha padrão e avisará o usuário para mudá-la para uma senha mais segura.

NOTA: *Escolhendo uma senha*: Ter uma senha segura é a primeira linha de defesa contra invasão. Você não quer ter uma senha de fácil dedução porque isto torna mais fácil alguém entrar em seu sistema. Uma senha ideal seria uma sequência aleatória de caracteres, incluindo letras maiúsculas e minúsculas, números e caracteres aleatórios. (Um caractere tab pode não ser uma boa escolha dependendo em qual tipo de computador você estará logando.) Há muitos pacotes de softwares que geram senha aleatórias para você; procure na Internet por esses

utilitários.

Em geral, use o bom senso: não coloque uma senha que seja a data de aniversário de alguém, algo encontrado em sua mesa, ou qualquer coisa facilmente associada com você. Uma senha como "seguro1"ou qualquer outra senha que você ver impresso ou online também é ruim.

Remover usuários não é uma tarefa complicada. Apenas execute *userdel* com o nome da conta para removê-la. Você deve verificar se o usuário não está logado e se não há processos do usuário em execução. E mais, lembre que uma vez excluido o usuário, toda a informação da senha do usuário terá sido apagada permanentemente.

```
# userdel jellyd
```

Este comando remove o usuário *jellyd* do seu sistema. O usuário é removido dos arquivos */etc/passwd*, */etc/shadow*, e */etc/group*, mas não remove o diretório home do usuário.

Se você deseja também remover o diretório home, você deve usar este comando:

```
# userdel -r jellyd
```

Desabilitar temporariamente uma conta será tratada na próxima seção sobre senhas, uma vez que uma mudança temporária envolve mudar a senha do usuário. Mudar outra informação da conta é tratada na seção 12.1.3.

Os programas para adicionar e remover grupos são bem simples. *groupadd* adicionará outra entrada no arquivos */etc/group* com um ID de grupo único, enquanto *groupdel* removerá o grupo especificado. É possível editar o arquivo */etc/group* para adicionar usuários em um grupo específico. Por exemplo, para adicionar um grupo chamado *cvs*:

```
# groupadd cvs
```

E removê-lo:

```
# groupdel cvs
```

12.1.2 Mudando senhas

O comando *passwd* muda as senhas modificando o arquivo */etc/shadow*. Este arquivo contém todas as senhas do sistema em formato criptografado. Para mudar sua senha, você digitaria:

```
$ passwd
Changing password for chris
Old password:
Enter the new password (minumum of 5, maximum of 127 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
```

Como se pode ver, você é solicitado para entrar com sua senha antiga. Ela não aparecerá na tela quando você digitá-la, o mesmo ocorre quando você se loga. Então, você é solicitado para entrar com a nova senha. *passwd* realiza várias verificações em sua nova senha e avisará se ela não passar nas verficações. Você pode ignorar esses avisos caso queira. Você será solicitado

para entrar com sua nova senha por uma segunda vez para confirmá-la.

Se você é *root*, você também pode mudar a senha de outros usuários:

```
# passwd ted
```

Você então passará pelo mesmo procedimento acima, com exceção que você não terá que digitar a senha antiga do usuário (um dos muitos benefícios de ser *root*...)

Se precisar, você também pode desabilitar temporiamente uma conta, e reabilitá-la mais tarde se for preciso. Desabilitar e reabilitar uma conta podem ser feitos com *passwd*. Para desabilitar uma conta, faça o seguinte como *root*:

```
# passwd -l david
```

Isto mudará a senha de david para algo que nunca coincida com qualquer valor criptografado. Você reabilita a conta usando:

```
# passwd -u david
```

Agora, a conta de david está de volta ao normal. Desabilitar uma conta pode ser útil se o usuário não segue as regras que você definiu para seu sistema, ou se ele exportou uma cópia muito grande de *xeyes*(1) para seu desktop X.

12.1.3 Mudando a Informação do Usuário

Há dois tipos de informação que os usuários podem mudar a qualquer hora: o shell e o finger. O Slackware Linux usa *chsh* (change shell) e *chfn* (change finger) para modificar esses valores.

Um usuário pode definir qualquer shell que está listado no arquivo */etc/shells*. Para a maioria das pessoas, */bin/bash* será adequado. Outros podem estar familiarizados com um shell encontrado nos seus sistemas, no trabalho ou na escola e desejam usar aquele que eles já conhecem. Para mudar seu shell, use *chsh*:

```
$ chsh
Password:
Changing the login shell for chris
Enter the new value, or press return for the default
Login Shell [/bin/bash]:
```

Depois de digitar sua senha, digite o caminho completo do novo shell. Primeiro, tenha certeza que ele está listado no arquivo */etc/shells*(5). O usuário *root* pode mudar o shell de qualquer usuário executando *chsh* utilizando o nome do usuário como argumento.

A informação de finger é a informação opcional como nome completo, números de telefone e de sala. Estas informações podem ser mudadas usando *chfn*, e segue o mesmo procedimento feito na criação da conta. Como sempre, o usuário *root* pode mudar a informação de finger de qualquer um.

12.2 Usuários e Grupos, o modo difícil

Naturalmente, é possível adicionar, modificar e remover usuários e grupos sem usar os scripts e programas que vêm com o Slackware. Não é muito difícil, embora depois de ler este processo, você provavelmente achará muito mais fácil usar os scripts. Entretanto, é importante conhecer como realmente a informação de sua senha é armazenada, no caso de você precisar recuperar esta informação e não ter as ferramentas do Slackware disponível.

Primeiro, nós adicionaremos um novo usuário nos arquivos */etc/passwd*(5), */etc/shadow*(5) e */etc/group*(5). O arquivo *passwd* contém algumas informações sobre os usuários do seu sistema, mas (bastante estranho) não têm as senhas deles. Isto não era assim, mas foi alterado tempos atrás por questões de segurança. O arquivo passwd deve ser legível para todos os usuários, mas você não deseja senhas criptografadas legíveis, pois intrusos poderiam usar as senhas criptografadas como ponto inicial para descriptografar as senhas dos usuários. Ao invés disto, as senhas criptografadas são mantidas no arquivo shadow, que é legível apenas para o root, e a senha de todos aparecem no arquivo *passwd* simplesmente com um "x". O arquivo *group* lista todos os grupos e quem está em cada um deles.

Você pode usar o comando *vipw* para editar o arquivo */etc/passwd* de modo seguro, e o comando *vigr* para editar o arquivo */etc/group* também de modo seguro. Use *vipw -s* para editar o arquivo */etc/shadow*. ("Modo seguro neste contexto significa que ninguém mais será capaz de modificar o arquivo que você está editando no momento. Se você é o único administrador do seu sistema, você provavelmente está em segurança, mas é melhor pegar bons hábitos no início.)

Vamos examinar o arquivo */etc/passwd* e ver como adicionar um novo usuário. Uma típica entrada em *passwd* se parece como isto:

```
chris:x:1000:100:Chris Lumens,Room 2,,:/home/chris:/bin/bash
```

Cada linha corresponde a um usuário e os campos de cada linha são separados pelo caracter dois pontos. Os campos são o nome de login, senha criptografada (um "x"para todos em um sistema Slackware, sendo que o Slackware usa senhas mascaradas), ID do usuário, ID do grupo, informações adicionais (separadas por vírgula), diretório home e o shell. Para adicionar um novo usuário manualmente, adicione uma nova linha no final do arquivo preenchendo as informações apropriadas.

As informações que você adicionar precisam atender alguns requisitos, ou seu novo usuário poderá ter problemas ao se logar. Primeiro, tenha certeza que o campo da senha é um x, e que o nome do usuário e seu ID sejam únicos. Atribua um grupo para o usuário, ou 100 (o grupo "users"no Slackware) ou seu grupo padrão (use o número do grupo, não o nome). Dê ao usuário um diretório home válido (que você criará mais tarde) e o shell (lembre-se, os shells válidos estão listados em */etc/shells*).

Em seguida, precisaremos adicionar uma entrada no arquivo /etc/shadow, que contém as senhas criptografadas. Uma entrada típica se parece com:

```
chris:\$1\$w9bsw/N9\$uwLr2bRER6YyBS.CAEp7R.:11055:0:99999:7:::
```

Novamente, cada linha corresponde a uma pessoa, com os campos delimitados pelo caracter dois pontos. Os campos são (em ordem) nome de login, senha criptografada, dias deste a época (01 de janeiro de 1970) que a senha foi mudada pela última vez, dias antes da senha poder ser alterada, dias depois que a senha deve ser alterada, dias que o usuário é notificado antes da

senha expirar, dias depois da senha expirar que a conta é desabilidada, dias deste a época que a conta está desabilidada e um campo reservado.

Como você pode ver, a maioria dos campos são para informações de validade de contas. Se você não está usando uma informação deste, você precisa apenas preencher uns poucos campos com alguns valores em especial. Caso contrário, você precisará calcular e tomar decisões antes de poder preencher esses campos. Para um novo usuário, apenas coloque algum lixo aleatório no campo de senha. Não se preocupe se a senha está correta agora, porque você irá mudá-la em um minuto. O único caractere que você não pode incluir no campo da senha é o dois pontos. Deixe o campo "dias desde que a senha foi alterada"em branco. Preencha com 0,99999, e 7 como você viu no exemplo acima e deixe os outros campos em branco..

(Para aqueles que pensam que vendo minha senha criptograda acima acreditam que têm uma brecha para entrar em meu sistema, então sigam em frente. Se você decifrar aquela senha, você saberá a senha para um teste de firewall do sistema. Isto é muito útil. :))

Todos os usuário comuns são membros do grupo "*users*"num sistema típicamente Slackware. Entretanto, se você deseja criar um novo grupo, ou adicionar o usuário em grupos adicionais, você precisará modificar o arquivo */etc/group*. Aqui está uma típica entrada:

```
cvs::102:chris,logan,david,root
```

Os campos são nome do grupo, senha do grupo, ID do grupo e membros do grupos, separados por vírgulas. Criar um novo grupo é simples como adicionar uma nova linha com um número de ID único e listar todos os usuários que você quer no grupo. Qualquer usuário que está no novo grupo e está logado terá que logar novamente para que as mudanças tenham efeito.

Neste ponto, pode ser uma boa idéia usar os comandos *pwck* e *grpck* para verificar se as mudanças que você fez são consistentes. Primeiro, use *pwck -r* e *grpck -r*: a opção *-r* não faz as alterações, mas lista as alterações que você teria feito se você executasse o comando sem aquela opção. Você pode usar a saída do comando para decidir se precisa modificar mais algum arquivo, executando *pwck* ou *grpck* sem a opção*-r*, ou simplesmente deixar as mudanças como estão.

Neste ponto, você deve usar o comando *passwd* para criar uma senha apropriada para o usuário. Então, use o comando *mkdir* para criar o diretório home do novo usuário no local que você entrou no arquivo */etc/passwd* e o comando *chown* para mudar o proprietário do novo diretório.

Remover um usuário é uma simples questão de excluir todas as entradas que existem do usuário. Remova a entrada do usuário dos arquivos */etc/passwd* e */etc/shadow*, e remova o nome de login dos grupos no arquivo */etc/group*. Se você desejar, apagar o diretório home, o arquivo de spool de mail (se eles existirem).

Remover grupos é similar: remova as entradas do arquivo */etc/group*.

12.3 Desligando o sistema apropriadamente

É muito importante que você desligue seu sistema apropriadamente. Simplesmente desligar pelo botão de energia pode causar sérios estragos no sistema de arquivos. Enquanto o sistema está ligado, arquivos estão em uso mesmo que você não esteja fazendo nada. Lembre-se que

há muitos processos rodando em segundo plano o tempo todo. Esses processos são gerenciados pelo sistema e mantêm vários arquivos abertos. Quando o sistema é desligado pelo botão de energia, esses arquivos não são fechados apropriadamente e podem ficar corrompidos. Dependendo de quais arquivos são corrompidos, o sistema pode ficar completamente inutilizado! Em qualquer caso, você terá que passar por um processo longo de checagem do sistema de arquivos na próxima inicialização.

NOTA: Se você configurou seu sistema com um sistema de arquivos com journalling, como ext3 ou reiserfs, você está parcialmente protegido contra estragos no sistema de arquivos, e a checagem do sistema de arquivos é mais curta se você usa um sistema de arquivo sem journalling, como ext2. Entretanto, esta segurança não é desculpa para desligar seu sistema de modo inapropriado! Um sistema de arquivos com journalling é uma tentativa de proteger seus arquivos de eventos além de seu controle, não contra sua própria indolência.

Em todo caso, quando você quiser reinicializar ou desligar seu computador, é importante fazê-lo corretamente. Há vários meios de se fazer isso; você pode usar o que achar mais divertido (ou menos trabalhoso). Sendo que desligar e reinicializar são procedimentos similares, a maioria dos meios de desligar o sistema também podem ser aplicados para reinicializá-lo.

O primeiro método é através do comando *shutdown*(8), e é provavelmente o mais popular. *shutdown* pode ser usado para reinicializar ou desligar o sistema em um dado tempo e pode mostrar uma mensagem a todos os usuários logados no sistema dizendo que está sendo desligado.

O modo mais básico de desligar o computador é:

```
# shutdown -h now
```

Neste caso, não será enviada uma mensagem personalizada para os usuários; eles verão a mensagem padrão de *shutdown.* ”*now*”é o tempo que você quer desligar e o ”*-h*”significa terminar o sistema. Não é um modo muito amigável de se fazer em um sistema multiusuários, mas funciona adequadamente em seu computador de casa. Um método melhor em um sistema multiusuário é dar a todos um aviso antecipado:

```
# shutdown -h +60
```

Isto desligará o sistema em uma hora(60 minutos), que é adequado em um sistema multiusuário normal. Sistemas essenciais devem ter a sua hora de desligar agendada antecipadamente, e você deve colocar avisos sobre a hora de desligar em quaisquer lugares usados apropriadamente para notificações do sistema (email, quadro de aviso, */etc/motd*, ou qualquer outro).

Para reinicializar o sistema, usa-se o mesmo comando, mas substitui-se ”*-h*”por ”*-r*”:

```
# shutdown -r now
```

Você pode usar a mesma notação de tempo com *shutdown -r* que você usuaria com *shutdown -h.* Há muitas outras coisas que você pode fazer com o comando *shutdown* para controlar quando desligar ou reinicializar a máquina; veja a página de manual para mais detalhes.

O segundo modo de desligar ou reinicializar o computador é usar os comandos *halt*(8) e *reboot*(8). Como os nomes indicam, *halt* terminará imediatamente a operação do sistema e *reboot* reinicializará o sistema. (*reboot* é na verdade apenas um link simbólico para o comando

halt.) Eles são executados como:

```
# halt
# reboot
```

Um meio de baixo nível de reinicializar ou desligar o sistema é usar diretamente o comando *init*. Todos os outros métodos são simplesmente meios convenientes de usar *init*, mas você pode dizer diretamente ao *init* o que fazer usando *telinit*(8) . Usar *telinit* dirá ao *init* que nível de execução (runlevel) acionar, o que fará um script especial ser executado. Este script matará ou ressusitará processos necessários para aquele nível de execução. Isto funciona para reinicializar e desligar porque ambos são níveis especiais de execução.

```
# telinit 0
```

O nível de execução 0 é o modo de desligar. Dizer ao *init* para entrar neste nível fará todos os processos serem terminados, o sistema de arquivos será desmontado e a máquina será desligada. Isto é um modo perfeitamente aceitável de desligar o sistema. Em muitos laptops e computadores desktops modernos, isso fará com que a máquina seja desligada.

```
# telinit 6
```

O nível de execução 6 é o modo de reinicialização. Todos os processos serão terminados, o sistema de arquivo será desmontado e a máquina será reinicializada. Isto é um modo perfeitamente aceitável de reinicializar o sistema.

Para os curiosos, quando muda-se para os níveis de execução 0 ou 6, seja usando *shutdown*, *halt*, ou *reboot*, o script */etc/rc.d/rc.6* é executado. (O script */etc/rc.d/rc.0* um link simbólico para */etc/rc.d/rc.6*.) Você pode personalizar este arquivo a seu gosto – mas esteja certo de testar suas mudanças cuidadosamente!

Há um último método de reinicializar o sistema. Todos os outros métodos requerem que você esteja logado como *root*. Entretanto, é possível reinicializar a máquina mesmo se você não é o adminstrador, contanto que você tenha acesso físico ao teclado. Teclar **Control+Alt+Delete** (a "saudação de três dedos") fará a máquina reinicializar imediatamente. (Por trás deste cenário, o comando *shutdown* é chamado para você quando usa **Control+Alt+Delete**.) Isto nem sempre funciona quando se está usando o X Windows – você pode precisar usar **Control+Alt+F1** (ou outra tecla de função) para alternar para um terminal em modo texto antes.

Finalmente, o arquivo que efetivamente controla cada aspecto da inicialização e do desligamento é o arquivo */etc/inittab*(5). Em geral, você não deve modificar este arquivo, mas ele pode dar a você noções de como algumas coisas funcionam. Como sempre, veja as páginas de manual para mais detalhes.

Capítulo 13

Comandos de rede básicos

Uma rede consiste em vários computadores conectados entre si. A rede pode ser tão simples quanto alguns computadores ligados em casa ou no escritório, ou tão complicada quanto uma rede de uma grande universidade ou a Internet. Quando seu computador é parte de uma rede, você tem acesso a esses sistemas diretamente ou através de serviços como o correio eletrônico ou a web.

Existe uma variadade de programas para redes que você pode usar. Alguns são úteis para realizar diagnósticos como verificar se a rede está funcionando corretamente. Outros (como leitores de e-mail e navegadores) são úteis para trabalhar ou manter contato com outras pessoas.

13.1 ping

O *ping*(8) envia um pacote ICMP *ECHO_REQUEST* para o host especificado. Se o host responde, você recebe o pacote ICMP de volta. Parece estranho? Bom, você pode "dar um ping"em um endereço IP para saber se a máquina está conectada. Se não obtiver resposta, você sabe que alguma coisa está errada.

Aqui está um exemplo de conversa entre dois usuários Linux:

```
Usuário A: A máquina Loki está desligada de novo.
Usuário B: Você tem certeza?
Usuário A: Sim, eu tentei pingá-la, mas não obtive resposta.
```

São circunstâncias como essas que fazem o *ping* ser um comando muito útil no dia a dia. Ele provê uma forma muito rápida de ver se uma máquina está ligada e conectada à rede. A sintaxe básica é:

```
$ ping www.slackware.com
```

Existem, obviamente, várias opções que podem ser especificadas. Olhe a página de manual de *ping(1)* para maiores informações.

13.2 traceroute

O *traceroute*(8) do Slackware é uma ferramenta de diagnóstico de rede muito útil. *traceroute* mostra por quais hosts um pacote passa até chegar ao seu destino. Você pode ver quantos hosts

tem do seu computador até o site do Slackware com esse comando:

```
$ traceroute www.slackware.com
```

Cada host vai ser mostrado, assim como o tempo de resposta de cada host. Aqui está um exemplo de saida do comando:

```
$ traceroute www.slackware.com
traceroute to www.slackware.com (204.216.27.13), 30 hops max, 40 byte packets
1  zuul.tdn (192.168.1.1)  0.409 ms  1.032 ms  0.303 ms
2  207.171.227.254 (207.171.227.254)  18.218 ms  32.873 ms  32.433 ms
3  border-sf-2-0-4.sirius.com (205.134.230.254) 15.662 ms 15.731 ms 16.142 ms
4  pb-nap.crl.net (198.32.128.20)  20.741 ms  23.672 ms  21.378 ms
5  E0-CRL-SFO-03-E0X0.US.CRL.NET (165.113.55.3) 22.293 ms 21.532 ms 21.29 ms
6  T1-CDROM-00-EX.US.CRL.NET (165.113.118.2)  24.544 ms  42.955 ms 58.443 ms
7  www.slackware.com (204.216.27.13)  38.115 ms  53.033 ms  48.328 ms
```

O *traceroute* é similar ao *ping*, já que o mesmo usa pacotes ICMP também. Existem várias opções que você pode especificar para o *traceroute*. Essas opções são explicadas em detalhes na respectiva página de manual.

13.3 Ferramentas DNS

O Domain Name Service (abreviado por DNS) é o protocolo mágico que permite a seu computador converter nomes de domínio como www.slackware.com em endereços IP 64.57.102.34. Computadores não podem rotear pacotes para www.slackware.com, mas eles podem rotear pacotes para o endereço IP desse domínio. Isso nos dá uma forma conveniente de lembrarmos de máquinas. Sem o DNS teríamos que manter um banco de dados mental de qual endereço IP pertence a qual computador, e assumindo que o endereço IP é sempre o mesmo. Claramente usar nomes para computadores é melhor, mas como nós mapeamos nomes em endereços IP?

13.3.1 host

O *host*(1) pode fazer isso por nós. *host* é usado para mapear nomes em endereços IP. É um utilitário muito rápido e simples, sem muitas funções.

```
$ host www.slackware.com
www.slackware.com is an alias for slackware.com.
slackware.com has address 64.57.102.34
```

Mas vamos supor que por alguma razão nós queremos mapear um endereço IP para um nome de domínio; e então?

13.3.2 nslookup

nslookup é um programa que tem resistido ao tempo. *nslookup* está depreciado e pode ser removido de lançamentos futuros. Não existe nem mesmo uma página de manual para este programa.

```
$ nslookup 64.57.102.34
Note:  nslookup is deprecated and may be removed from future releases.
Consider using the 'dig' or 'host' programs instead.  Run nslookup with
the '-sil[ent]' option to prevent this message from appearing.
Server:   192.168.1.254
Address:  192.168.1.254#53

Non-authoritative answer:
www.slackware.com canonical name = slackware.com.
Name:   slackware.com
Address: 64.57.102.34
```

13.3.3 dig

O Domain Information Groper, *dig*(1) abreviando, é o programa que pega informações DNS. *dig* pode pegar qualquer informação de um servidor DNS incluindo reverse lookups, A, CNAME, MS, SP, e registros TXT. *dig* possui muitas opções de linha de comando e se você não é familiar com eles, leia a extensiva página de manual da ferramenta.

```
$ dig @192.168.1.254 www.slackware.com mx

; <<>> DiG 9.2.2 <<>> @192.168.1.254 www.slackware.com mx
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26362
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.slackware.com.       IN      MX

;; ANSWER SECTION:
www.slackware.com.76634   IN      CNAME   slackware.com.
slackware.com.    86400   IN      MX      1 mail.slackware.com.

;; AUTHORITY SECTION:
slackware.com.    86400   IN      NS      ns1.cwo.com.
slackware.com.    86400   IN      NS      ns2.cwo.com.

;; ADDITIONAL SECTION:
ns1.cwo.com.      163033  IN      A       64.57.100.2
ns2.cwo.com.      163033  IN      A       64.57.100.3

;; Query time: 149 msec
;; SERVER: 192.168.1.254#53(192.168.1.254)
;; WHEN: Sat Nov  6 16:59:31 2004
;; MSG SIZE  rcvd: 159
```

O exemplo acima deve dar uma idéia de como o *dig* funciona. "@192.168.1.254"especifica o servidor dns a ser usado. "www.slackware.com"é o nome de domínio que eu estou verificando, e "mx"é o tipo de verificação que eu estou fazendo. A query acima me diz que quando enviar um e-mail para www.slackware.com ele será enviado na verdade, para mail.slackware.com.

13.4 finger

O *finger*(1) irá retornar informações sobre o usuário especificado. Você utiliza o *finger* em um username ou em um endereço de e-mail e ele irá tentar contactar o servidor necessário e retornar o username, escritório, número de telefone, e outras informações. Aqui está um exemplo:

```
$ finger johnc@idsoftware.com
```

O *finger* pode retornar o username, status de e-mail, número de telefones, e o conteúdo dos arquivos ".plan"e ".project". Claro que a informação retornada varia conforme cada servidor *finger*. Aquele que está incluso no Slackware retorna as seguintes informações por padrão:

- Username
- Room number
- Home phone number
- Work phone number
- Login status
- Email status
- Conteúdo do arquivo *.plan* dentro do diretório home do usuário
- Conteúdo do arquivo *.project* dentro do diretório home do usuário

Os primeiros quatro itens podem ser configurados com o comando *chfn*. Ele guarda essas informações no arquivo */etc/passwd*. Para mudar a informação no seu arquivo *.plan* ou *.project*, basta editar esses arquivos com seu editor de texto favorito. Esses arquivos precisam estar no seu diretório home e precisam ser chamados *.plan* e *.project* obrigatoriamente.

Muitos usuários utilizam *finger* na sua própria conta de uma máquina remota para ver rapidamente se possuem novos e-mails. Você também pode utilizá-lo para ver os planos de um usuário ou o seu projeto atual.

Como muitos comandos, *finger* possui opções. Olhe a página de manual dele para maiores informações sobre opções que você pode usar.

13.5 telnet

Alguém uma vez disse que o *telnet*(1) tinha sido a coisa mais legal que já havia visto em computadores. A habilidade de logar remotamente e realizar tarefas em outro computador é o que separa o Unix e sistemas operacionais "Unix-like"de outros sistemas operacionais.

O *telnet* permite que você logue em um computador, assim como se você estivesse sentado em frente ao mesmo. Depois de ter seu usuário e senha verificados, você terá um prompt do shell. Depois disso, você pode fazer qualquer coisa que um console em modo texto permitir. Compor e-mail, ler grupo de notícias, mover arquivos, e assim por diante. Se você estiver rodando o X e fizer *telnet* para outra máquina, você pode rodar aplicativos do X no computador remoto e mostrar o resultado no seu computador.

Para logar em uma máquina remota, use a seguinte sintaxe:

```
$ telnet <hostname>
```

Se o host responder, você receberá o prompt de login. Digite seu usuário e sua senha. Pronto. Você agora possui uma shell. Para sair da sua sessão telnet, use o comando *exit* ou o comando *logout*.

AVISO: *telnet*não encripta a informação que transmite. Tudo é transmitido em texto puro, até mesmo senhas. Não é recomendado usar *telnet* através da Internet. Para tal, considere usar o *ssh* (Secure Shell). Ele encripta todo o tráfego e está disponível gratuitamente.

13.5.1 Outras utilidades do telnet

Agora que nós convencemos você a não utilizar o protocolo telnet para logar em uma máquina remota, nós iremos mostrar algumas outras utilidades do *telnet*.

Você também pode usar o *telnet* para conectar em um host em uma certa porta.

```
$ telnet <hostname> [port]
```

Isso pode ser bem útil quando você precisa rapidamente testar um certo serviço, e você precisa de total controle sobre os comandos, além de ver precisamente o que está acontecendo. Você pode testar interativamente ou usar um servidor SMTP, um servidor POP3, um servidor HTTP, etc. dessa forma.

Na próxima figura você verá como você pode usar *telnet* em um servidor HTTP na porta 80, e conseguir algumas informações básicas dele:

Usando telnet em um servidor web

```
$ telnet store.slackware.com 80
Trying 69.50.233.153...
Connected to store.slackware.com.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 25 Apr 2005 20:47:01 GMT
Server: Apache/1.3.33 (Unix) mod_ssl/2.8.22 OpenSSL/0.9.7d
Last-Modified: Fri, 18 Apr 2003 10:58:54 GMT
ETag: "193424-c0-3e9fda6e"
Accept-Ranges: bytes
Content-Length: 192
Connection: close
Content-Type: text/html

Connection closed by foreign host.
$
```

Você pode fazer a mesma coisa para qualquer outro protocolo que seja de texto puro, desde que saiba em que porta conectar e quais são seus comandos.

13.6 O ssh (Secure Shell)

Hoje, o *ssh* desfruta da adoração que o telnet desfrutou uma vez. *ssh*(1) nos permite realizar conexões em uma máquina remota e executar programas como se você estivesse fisicamente presente; contudo, o *ssh* encripta todos os dados transferidos entre os dois computadores. Dessa forma, mesmo que outras pessoas interceptem a comunicação, eles não serão capazes de entendê-la. Uma conexão ssh típica está abaixo:

```
$ ssh carrier.lizella.net -l alan
The authenticity of host 'carrier.lizella.net (192.168.1.253)' can't be
established.
RSA key fingerprint is 0b:e2:5d:43:4c:39:4f:8c:b9:85:db:b2:fa:25:e9:9d.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'carrier.lizella.net' (RSA) to the list of
known hosts.
Password: password
Last login: Sat Nov  6 16:32:19 2004 from 192.168.1.102
Linux 2.4.26-smp.
alan@carrier:~$ ls -l MANIFEST
-rw-r--r--  1 alan users 23545276 2004-10-28 20:04 MANIFEST
alan@carrier:~$ exit
logout
Connection to carrier.lizella.net closed.
```

Acima você pode perceber que eu estou fazendo uma conexão ssh em carrier.lizella.net, e checando as permissões do arquivo *MANIFEST*.

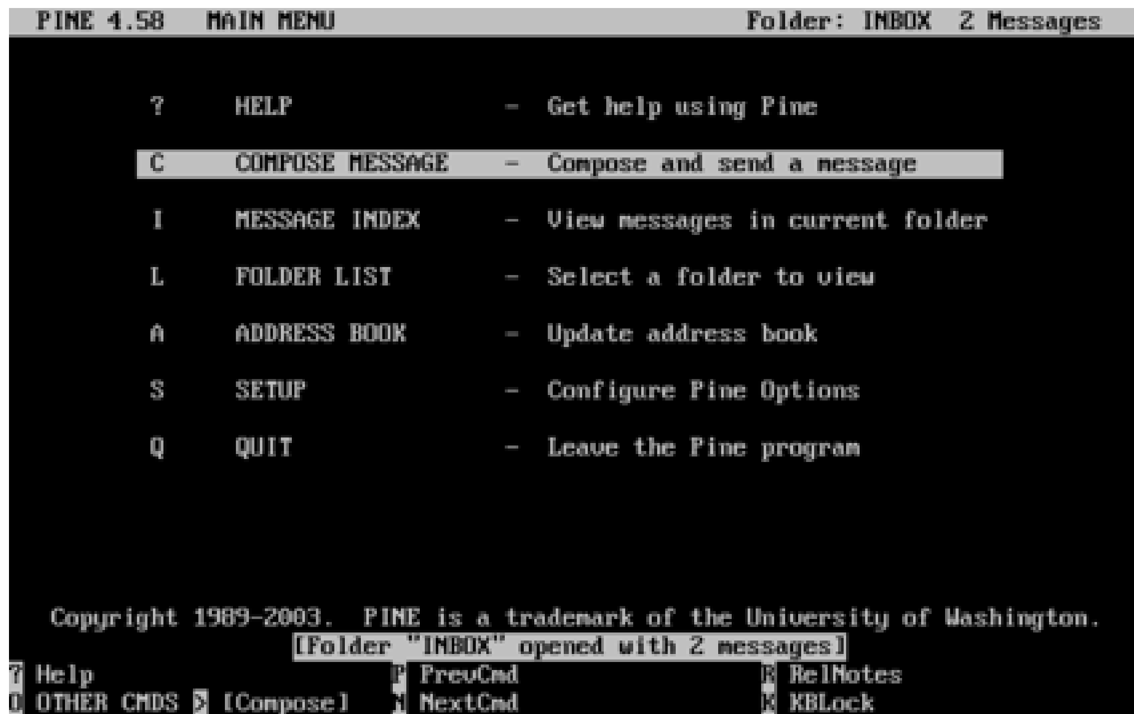
13.7 email

O correio eletrônico é uma das coisas mais populares que você pode fazer na Internet. Em 1998 descobriu-se que mensagens eletrônicas eram mais utilizadas que o correio normal. Ele é certamente muito comum e útil.

No Slackware, nós fornecemos um servidor de email padrão, e muitos clientes de email. Todos os clientes discutidos abaixo são utilizados em modo texto. Muitos dos usuários Windows podem ser contra isso, mas você descobrirá que um cliente utilizado em modo texto é muito conveniente, especialmente quando você está checando email remotamente. Não tema pois existem diversos clientes gráficos como o Kmail do KDE. Se você deseja usar um desses, verifique o menu de ajuda dos mesmos.

13.7.1 pine

O *pine*(1) não é o *elm*. A Universidade de Washington criou o seu programa para ler emails e notícias da Internet e ao mesmo tempo ser um leitor fácil para seus estudantes. O *pine* é um dos clientes de email mais populares hoje em uso e está disponível para todos os sabores de Unix e até mesmo Windows.



Você verá um menu de comandos e uma linha de teclas de comando na parte de baixo da tela. Contudo o *pine* é um programa complexo, e nós não iremos discutir todos os seus recursos neste texto.

Para ver o que está na sua caixa de entrada, digite i. Suas mensagens são listadas com suas datas, autores e assuntos. Selecione a mensagem que você deseja e aperte enter para visualizá-la. Digitando r você poderá responder a mensagem. Após digitar a mensagem de resposta, digite **Ctrl+X** para enviá-la. Você pode digitar i para voltar à lista de mensagens.

Se você deseja excluir a mensagem, digite d. Isso irá marcar a mensagem selecionada para exclusão. O *pine* apaga as mensagens quando você sai do programa. O *pine* também permite que você guarde seus emails em pastas. Você pode obter a listagem das pastas pressionando a tecla l. Na listagem de mensagens, pressione s para salvá-la em outra pasta. Ele irá perguntar em qual pasta deve salvar a mensagem.

O *pine* oferece muitas, muitas funcionalidades; você definitivamente deve dar uma olhada na página de manual para maiores informações. Ele irá conter as últimas informações sobre o programa.

13.7.2 elm

elm(1) é outro cliente de email em modo texto bastante popular. Apesar de não possuir uma interface tão amigável quanto o *pine*, está por aí há muito mais tempo.

```
          Mailbox is '/var/mail/root' with 2 messages [ELM 2.5 PL6]

     1   Sep 17   Patrick J. Volkerd (174)  Welcome to Linux (Slackware 9.1)!
  O  2   Sep 17                      (45)   Register with the Linux counter pr

     You can use any of the following commands by pressing the first character:
   d)elete or u)ndelete mail,  m)ail a message,  r)eply or f)orward mail,  q)uit
      To read a message, press <return>.  j = move down, k = move up, ? = help

Command:
```

Por padrão você está na sua caixa de entrada. As mensagens são listadas com o número da mensagem, data, autor e assunto. Use as setas para selecionar a mensagem que deseja. Pressione Enter para ler a mensagem.

Para compor uma nova mensagem, pressione m na tela principal. A tecla d irá marcar a mensagem para exclusão. E a tecla r irá responder a mensagem que você está atualmente lendo. Todas essas teclas são mostradas na parte de baixo da tela com um prompt.

A página de manual apresenta o *elm* mais detalhadamente, então você provavelmente irá consultá-lo antes de utilizar o *elm*.

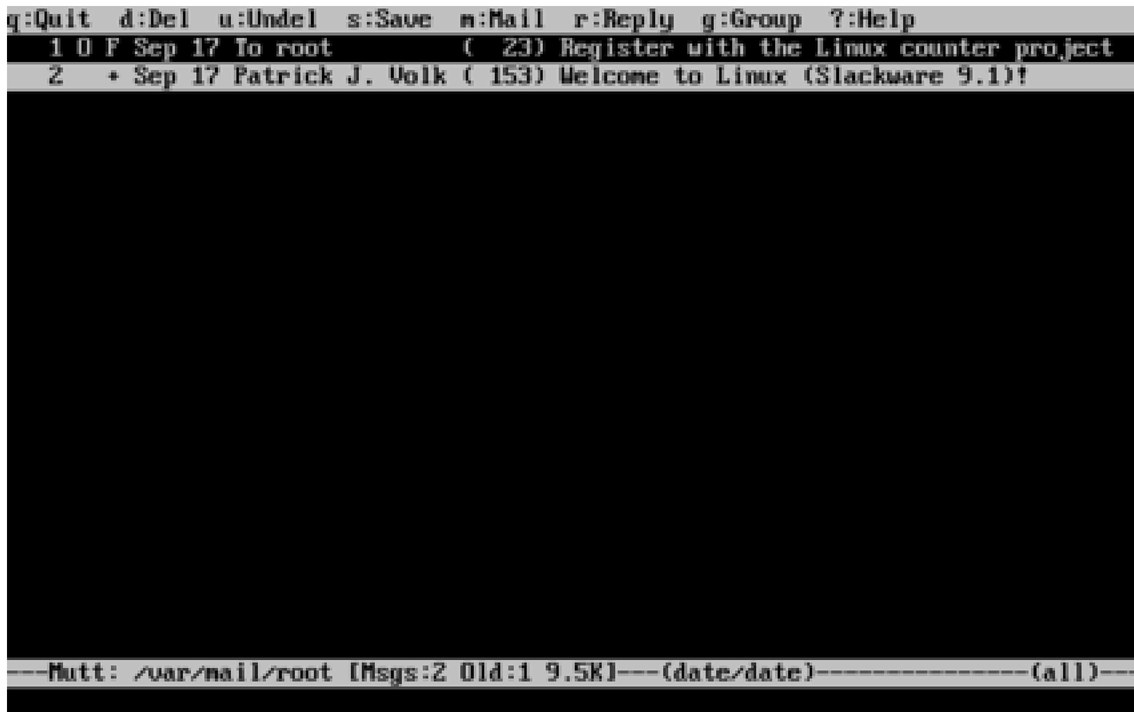
13.7.3 mutt

"Todos os clientes de email são ruins. Este é apenas menos ruim". A interface original do *mutt* era baseada na interface do *elm* com recursos adicionais encontrados em outros clientes de email populares, resultando em um *mutt* híbrido.

Alguns dos recursos do *mutt* incluem:

- Suporte a cores
- Agrupamento de mensagens
- Suporte a MIME e PGP/MIME
- Suporte a pop3 e imap
- Suporte a vários formatos de caixa de correio (mbox, MMDF, MH, maildir)
- *Altamente* personalizável

Tela principal do Mutt



Se você está procurando por um cliente de email que irá permitir que você tenha total controle sobre tudo, então você irá gostar do *mutt*. Todas as configurações padrão podem ser customizadas, associações de teclas mudadas. Se você quiser adicionar uma macro, você também pode.

Você provavelmente irá querer dar uma olhada na página de manual do *muttrc*, que irá explicar como configurar tudo isso. Ou dê uma olhada no exemplo de arquivo de configuração *muttrc*.

13.7.4 nail

nail(1) é um cliente de email de linha de comando. É muito primitivo e oferece quase nenhum recurso no que diz respeito à interface com usuário. Contudo, mailx é útil para aquelas vezes em que você precisa rapidamente enviar alguma coisa, testando sua instalação de MTA ou qualquer coisa parecida. Note que o Slackware cria um link simbólico para o *nail* em */usr/bin/mail* e */usr/bin/mailx*. Qualquer um desses três comandos executa o mesmo programa. De fato, você verá normalmente o *nail* sendo referenciado como *mail*.

A linha de comando básica é:

```
$ mailx <subject> <to-addr>
```

mailx lê o corpo da mensagem da entrada padrão. Então você pode usar o cat em um arquivo como entrada padrão para este comando para enviá-lo, ou você pode apenas digitar o texto e apertar **Ctrl+D** quando tiver acabado a mensagem.

Aqui está um exemplo de enviar por email o código fonte de um programa para outra pessoa:

```
$ cat randomfunc.c | mail -s "Here's that function" asdf@example.net
```

A página de manual explica mais do que o *nail* pode fazer, então você provavelmente irá querer dar uma olhada nele antes de usá-lo.

13.8 Navegadores

A primeira coisa que as pessoas pensam quando ouvem a palavra Internet é ”navegar na internet”. Em outras palavras, navegar em sites usando um navegador web. Esse é provavelmente de longe o uso mais popular da Internet para o usuário médio.

O Slackware fornece navegadores web que são gráficos na série ”XAP”, bem como navegadores web em modo texto na série ”N”. Nós iremos dar uma rápida olhada nas opções mais comuns.

13.8.1 lynx

lynx(1) é um navegador web em modo texto. É uma forma bem rápida de olhar alguma coisa na Internet.

Para iniciar o *lynx*, apenas digite *lynx* no prompt:

```
$ lynx
```

Página inicial padrão do Lynx

```
                                                            Lynx Information
                                     Lynx

  Lynx is a text browser for the World Wide Web.     Lynx 2.8.3 runs on
  Un*x, VMS, Windows 95/98/NT but not 3.1 or 3.11, on DOS (386 or
  higher) and OS/2 EMX.  The  current developmental version is also
  available for testing.  Ports to Mac are in beta test.

    * Many user questions are answered in the  online help provided with
      Lynx. Press the '?' key to find this help.
    * If you are encountering difficulty with Lynx you may write to
      lynx-dev@sig.net.  Be as detailed as you can about the URL where
      you were on the Web when you had trouble, what you did, what Lynx
      version you have (try '=' key), and what OS you have.  If you are
      using an older version, you may well need to upgrade.

  Maintained by  lynxdev@browser.org.

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-' to go back.
  Arrow keys: Up and Down to move.  Right to follow a link; Left to go back.
 H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

Você pode querer especificar um site para o *lynx* abrir:

```
$ lynx www.slackware.com
```

lynx exibe as teclas de comando e o que cada tecla faz na parte de baixo da tela. As setas para cima e para baixo movem através do documento, Enter seleciona o link destacado, e a seta da esquerda volta à página anterior. Digitando d irá fazer o download do arquivo selecionado. O comando g traz o prompt de endereço, para você digitar uma URL para o lynx abrir.

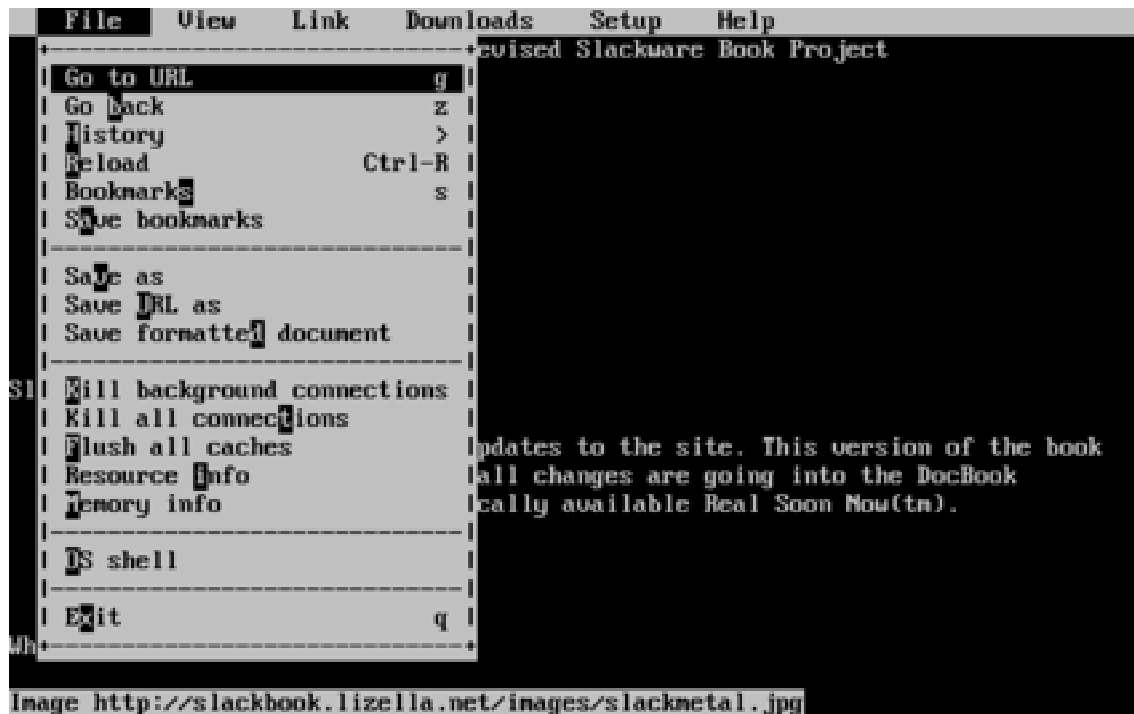
Existem muitos outros comandos no *lynx*. Você pode consultar a página de manual ou digitar h para que a tela de ajuda apareça para maiores informações.

13.8.2 links

Assim como no *lynx*, o *links* é um navegador em modo texto, onde você navega utilizando o teclado. Contudo, quando pressiona a tecla Esc, irá ativar um menu muito conveniente no

topo da tela. Isso permite que o *links* seja muito fácil de usar, sem ter que aprender um monte de teclas de atalho. Pessoas que não utilizam navegadores em modo texto todo dia irão gostar deste recurso.

O *links* parece ter um suporte melhor para frames e tabelas, quando comparado ao *lynx*.



13.8.3 wget

wget(1) é uma ferramenta de linha de comando que irá fazer o download de arquivos de uma URL específica. Como não é na verdade um navegador web, o *wget* é usado principalmente para fazer o download de sites inteiros ou parte deles para visualização offline dos mesmos, ou para o download de arquivos de servidores HTTP ou FTP. A sintaxe básica é:

```
$ wget <url>
```

Você também pode passar opções. Por exemplo, a opção abaixo irá fazer o download do site do Slackware:

```
$ wget --recursive www.slackware.com
```

O *wget* irá criar um diretório chamado www.slackware.com e irá guardar os arquivos dentro dele, assim como o site o faz.

O *wget* também pode fazer o download de sites FTP; apenas especifique uma URL FTP ao invés de uma HTTP.

```
$ wget ftp://ftp.gnu.org/gnu/wget/wget-1.8.2.tar.gz
--12:18:16--  ftp://ftp.gnu.org/gnu/wget/wget-1.8.2.tar.gz
     => ‘wget-1.8.2.tar.gz’
Resolving ftp.gnu.org... done.
Connecting to ftp.gnu.org[199.232.41.7]:21... connected.
```

```
Logging in as anonymous ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done. ==> CWD /gnu/wget ... done.
==> PORT ... done.    ==> RETR wget-1.8.2.tar.gz ... done.
Length: 1,154,648 (unauthoritative)

100%[=================================>] 1,154,648     209.55K/s    ETA 00:00

12:18:23 (209.55KB/s) - ‘wget-1.8.2.tar.gz’ saved [1154648]
```

O *wget* tem muitas outras opções, o que o torna muito bom para scripts específicos para sites (como espelhamento e assim por diante). A página de manual deve ser consultado para mais informações.

13.9 Clientes FTP

O FTP é o protocolo de transferência de arquivos. Ele permite que você envie e receba arquivos entre dois computadores. Existe o servidor FTP e o cliente FTP. Nós iremos falar sobre os clientes nesta seção. Para os curiosos, o ”cliente”é você. O ”servidor”é o computador que responde suas requisições FTP e aceita o seu login. Você irá fazer o download de arquivos do servidor e irá fazer upload de arquivos para o servidor. O cliente não pode aceitar conexões FTP, ele pode apenas conectar à servidores FTP.

13.9.1 ftp

Para conectar em um servidor FTP, simplesmente execute o comando *ftp*(1) e especifique o host:

```
$ ftp $<$hostname$>$ [port]
```

Se o host estiver rodando um servidor FTP, ele irá pedir um username e um password. Você poderá logar como você mesmo ou como ”anônimo”(anonymous). Sites FTP anônimos são bastante populares para repositório de arquivos. Por exemplo, para fazer o download do Slackware Linux por FTP, você deve usar FTP anônimo.

Uma vez conectado, você estará no prompt *ftp*>. Existem comandos especiais para FTP, mas eles são similares aos comandos padrão. A lista seguinte mostra alguns dos comandos básicos e o que eles fazem:

Comandos *ftp*

Comando	**Função**
ls	Lista arquivos
cd <diretório>	Muda de diretório
bin	Define modo de transferência binário
ascii	Define modo de transferência ASCII
get <arquivo>	Faz o download de um arquivo
put <arquivo>	Faz o upload de um arquivo
hash	Liga/Desliga a impressão de "#"para cada buffer transferido
tick	Mostra/Esconde contador de bytes
prom	Entra/Sai modo interativo para downloads
mget <arquivos>	Faz o download de um arquivo ou um grupo de arquivos; coringas são permitidos
mput <arquivos>	Faz o upload de um arquivo ou um grupo de arquivos; coringas são permitidos
quit	Sai do servidor FTP

Você também pode usar os seguintes comandos, que são auto-explicativos: *chmod*, *delete*, *rename*, *rmdir*. Para uma lista completa de todos os comandos e seus significados, apenas digite help ou ? e você verá a lista completa na sua tela.

FTP é um programa razoavelmente simples de ser usado, mas falta a interface que muitos de nós estão acostumados hoje em dia. A página de manual discute algumas das opções de linha de comando do *ftp*(1).

```
ftp> ls *.TXT
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
-rw-r--r--   1 root     100   18606 Apr  6  2002 BOOTING.TXT
-rw-r--r--   1 root     100   10518 Jun 13  2002 COPYRIGHT.TXT
-rw-r--r--   1 root     100     602 Apr  6  2002 CRYPTO_NOTICE.TXT
-rw-r--r--   1 root     100   32431 Sep 29 02:56 FAQ.TXT
-rw-r--r--   1 root     100  499784 Mar  3 19:29 FILELIST.TXT
-rw-r--r--   1 root     100  241099 Mar  3 19:12 PACKAGES.TXT
-rw-r--r--   1 root     100   12339 Jun 19  2002 README81.TXT
-rw-r--r--   1 root     100   14826 Jun 17  2002 SPEAKUP_DOCS.TXT
-rw-r--r--   1 root     100   15434 Jun 17  2002 SPEAK_INSTALL.TXT
-rw-r--r--   1 root     100    2876 Jun 17  2002 UPGRADE.TXT
226 Transfer complete.
ftp> tick
Tick counter printing on (10240 bytes/tick increment).
ftp> get README81.TXT
local: README81.TXT remote: README81.TXT
200 PORT command successful.
150 Opening BINARY mode data connection for README81.TXT (12339 bytes).
Bytes transferred: 12339
226 Transfer complete.
12339 bytes received in 0.208 secs (58 Kbytes/sec)
```

13.9.2 ncftp

O *ncftp*(1) (pronunciado "Nik-F-T-P") é uma alternativa ao tradicional cliente ftp que vem com o Slackware. Ele é ainda um programa em modo texto, mas oferece algumas vantagens em relação ao *ftp*, incluindo:

- ”Tab completion”
- Arquivo de favoritos(Bookmark)
- Uso mais liberal de coringas
- Histórico de comandos

Por padrão, *ncftp* irá tentar logar anonimamente ao servidor que você especificar. Você pode forçar o *ncftp* a apresentar o prompt de login com a opção ”*-u*”. Uma vez no sistema, você pode usar os mesmos comandos usados no *ftp*, você irá apenas notar uma interface melhor, que se parece mais com o *bash*.

```
ncftp /pub/linux/slackware > cd slackware-current/
Please read the file README81.TXT
  it was last modified on Wed Jun 19 16:24:21 2002 - 258 days ago
CWD command successful.
ncftp ...ware/slackware-current > ls
BOOTING.TXT         FAQ.TXT                  bootdisks/
CHECKSUMS           FILELIST.TXT             extra/
CHECKSUMS.asc       GPG-KEY                  isolinux/
CHECKSUMS.md5       PACKAGES.TXT             kernels/
CHECKSUMS.md5.asc   PRERELEASE_NOTES         pasture/
COPYING             README81.TXT             rootdisks/
COPYRIGHT.TXT       SPEEKUP_DOCS.TXT         slackware/
CRYPTO_NOTICE.TXT   SPEEK_INSTALL.TXT        source/
CURRENT.WARNING     Slackware-HOWTO
ChangeLog.txt       UPGRADE.TXT
ncftp ...ware/slackware-current > get README81.TXT
README81.TXT:                                   12.29 kB  307.07 kB/s
```

13.10 Conversando com outras pessoas

13.10.1 wall

wall(1) é uma forma rápida de escrever uma mensagem para os usuários em um sistema. A sintaxe básica é:

```
$ wall [file]
```

Isso irá fazer com que o conteúdo do arquivo seja mostrado no terminal de todos os usuários atualmente logados no sistema. Se você não especificar uma mensagem, *wall* irá ler da entrada padrão, então você pode simplesmente digitar sua mensagem e no final digitar **Ctrl+d**.

wall não possui muitos recursos, tirando o fato de servir para avisar a seus usuários que você está prestes a fazer alguma manutenção séria no sistema, ou até mesmo reiniciá-lo, para que eles salvem seus trabalhos e façam logoff =)

13.10.2 talk

O *talk*(1) permite que dois usuários conversem entre si. Ele divide a tela no meio, horizontalmente. Para requisitar um chat com outro usuário, use o comando:

```
$ talk <pessoa> [ttyname]
```



Se você especificar apenas o usuário, o *talk* assume que a requisição é local, então apenas usuários locais são requisitados. O nome do terminal é requerido se você deseja chamar um usuário em um terminal específico (se o usuário está logado em mais de um). A informação requerida para a conversa pode ser obtida com o comando *w*(1).

O *talk* também pode chamar usuários em hosts remotos. Para o username você simplesmente especifica um endereço de email. *talk* irá tentar contactar aquele usuário remoto naquele host.

talk é um pouco limitado. Ele somente suporta dois usuários e é half-duplex.

13.10.3 ytalk

ytalk(1) é uma reposição do *talk*. Ele vem com o Slackware como sendo o comando *ytalk*. A sintaxe é similar, mas existem algumas diferenças:

```
$ ytalk <username>[#ttyname]
```

```
==========================ˇ= YTalk version 3.1.1 =============================
test



--------------------------= root@192.168.1.132#tty1 =-------------------------
this is a test
```

O usuário e o terminal são especificados da mesma forma que no talk, com a diferença que você deve colocá-los juntos com o caractere (#).

ytalk oferece várias vantagens:

- Ele suporta mais de dois usuários.
- Um menu de opções pode ser ativado a qualquer momento com a tecla Esc.
- Você pode utilizar o shell enquanto estiver em uma conversa.
- E muito mais...

Se você é um administrador de servidores, você deve se certificar que a porta do *ntalk* está habilitada em */etc/inetd.conf*. *ytalk* precisa disso para funcionar corretamente.

Capítulo 14

Segurança

Segurança em um sistema é uma característica muito importante: ela pode prevenir que pessoas ataquem sua máquina, e também proteger dados sigilosos. Esse capítulo inteiro é dedicado a como proteger sua máquina Slackware contra script kiddies, crackers e outros ladrões de informações. Tenha em mente de que esse é só o começo de como proteger um sistema; segurança é um processo, não um estado.

14.1 Desabilitando Serviços

O primeiro passo após instalar o Slackware deve ser desabilitar todos os serviços que você não vai precisar. Qualquer serviço pode ser um risco de segurança, então é importante deixar habilitado o mínimo de serviços possíveis (ou seja, só os que são necessários). Os serviços são iniciados a partir de duas ferramentas principais o *inetd* e os scripts de inicialização (init scripts).

14.1.1 Serviços iniciados com o inetd

Muitos daemons que vem com o Slackware são iniciados a partir do *inetd*(8). O *inetd* é um daemon que escuta em todas as portas usadas pelos serviços configurados para serem iniciados por ele e inicia uma estância do daemon relevante quando uma tentativa de conexão é feita. Daemons iniciados pelo *inetd* podem ser desabilitados comentando-se a linha referente ao mesmo no arquivo */etc/inetd.conf*. Para fazer isso, abra esse arquivo no seu editor de textos favorito (ex. *vi*) e você verá linhas similares a essa:

```
$telnet stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

Você pode desabilitar esse serviço, e qualquer outro que não seja necessário para você, comentado-os (ou seja, incluindo um # (sustenido) ao início da linha). A linha acima ficaria assim:

```
#telnet stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

Depois que o *inetd* for reiniciado, esse serviço estará desabilitado. Você pode reiniciar o *inetd* com o comando:

```
# kill -HUP $(cat /var/run/inetd.pid)
```

14.1.2 Serviços iniciados pelos scripts de inicialização (init scripts)

Os outros serviços iniciados durante o boot da máquina são iniciados a partir dos scripts contidos em */etc/rc.d/*. Esses podem ser desabilitados de duas maneiras: a primeira seria remover a permissão de execução do script de inicialização desejado e a segunda seria comentar as linhas relevantes no script de inicialização.

Por exemplo, o SSH é iniciado pelo seu próprio script */etc/rc.d/rc.sshd*. Você pode desabilitá-lo usando:

```
# chmod -x /etc/rc.d/rc.sshd
```

Para os serviços que não possuem seu próprio script de inicialização, você precisará comentar as linhas relativas a eles em algum outro script de inicialização para desabilitá-los. Por exemplo, o daemon portmap é iniciado pelas seguintes linhas do arquivo */etc/rc.d/rc.inet2*:

```
# This must be running in order to mount NFS volumes.
# Start the RPC portmapper:
if [ -x /sbin/rpc.portmap ]; then
  echo "Starting RPC portmapper:  /sbin/rpc.portmap"
  /sbin/rpc.portmap
fi
# Done starting the RPC portmapper.
```

Esse serviço pode ser desabilitado adicionando um # ao inicio das linhas que ainda não começam com #, assim:

```
# This must be running in order to mount NFS volumes.
# Start the RPC portmapper:
#if [ -x /sbin/rpc.portmap ]; then
#  echo "Starting RPC portmapper:  /sbin/rpc.portmap"
#  /sbin/rpc.portmap
#fi
# Done starting the RPC portmapper.
```

Essas mudanças apenas terão efeito após um reboot ou mudando para o runlevel 3 ou 4. Você pode fazer isso digitando o seguinte em um console (você precisará se logar novamente após mudar para o runlevel 1):

```
# telinit 1
# telinit 3
```

14.2 Controle de Acesso ao Host

14.2.1 iptables

O *iptables* é o programa de configuração de filtros de pacotes para o Linux 2.4 e superior. O kernel 2.4 (2.4.5, para ser exato) foi introduzido no Slackware (como uma opção) na versão 8.0 e tornou-se o padrão no Slackware 8.1. Essa seção cobre apenas o básico de sua utilização e você pode acessar o site http://www.netfilter.org/ para mais detalhes. Esses comandos podem ser colocados no arquivo */etc/rc.d/rc.firewall*, o qual deve ter permissão de execução para que essas regras tenham efeito na inicialização. Note que comandos incorretos do *iptables* podem até mesmo te impedir de acessar sua própria máquina. A menos que você esteja 100% confiante de suas habilidades, sempre tenha certeza de que você tem acesso local a máquina.

A primeira coisa que a maior parte das pessoas devem fazer é configurar a politica padrão para cada método de entrada para DROP:

```
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
```

Quando tudo é barrado, você pode começar a liberar coisas. A primeira coisa a liberar é o trafego para seções que já estavam estabelecidas:

```
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT$
```

E para não impedir qualquer aplicação que use o endereço de loopback, é inteligente adicionar a seguinte regra:

```
# iptables -A INPUT -s 127.0.0.0/8 -d 127.0.0.0/8 -i lo -j ACCEPT$
```

Essa regra permite qualquer tráfego para e a partir do endereço 127.0.0.0/8 (127.0.0.0 - 127.255.255.255) na interface de loopback (*lo*). Quando for criar regras, é uma boa idéia ser o mais específico possível, para ter cereza de que suas regras não vão permitir inadvertidamente algo malicioso. Dito isso, regras que permitem pouco resultam em mais regras e mais digitação.

O próximo passo seria permitir o acesso a serviços específicos que rodem em sua máquina. Se, por exemplo, você quisesse rodar um servidor web na sua máquina, você usaria uma regra similar a essa:

```
# iptables -A INPUT -p tcp --dport 80 -i ppp0 -j ACCEPT$
```

Isso permitirá o acesso de qualquer máquina à porta 80 da sua máquina através da interface *ppp0*. Você pode querer restringir o acesso a esse serviço para que apenas certas máquinas possam acessá-lo. Essa regra permite o acesso ao seu servidor web a partir do endereço 64.57.102.34:

```
# iptables -A INPUT -p tcp -s 64.57.102.34 --dport 80 -i ppp0 -j ACCEPT$
```

Permitir o tráfego ICMP pode ser útil para efeitos de diagnóstico. Para fazer isso, você usuaria uma regra como essa:

```
# iptables -A INPUT -p icmp -j ACCEPT$
```

Muitas pessoas também vão querer configurar o serviço de Tradução de Endereços de Rede (Network Address Translation - NAT) em suas máquinas que atuem como gateway, para que outras máquinas na sua rede possam acessar a Internet através dela. Você pode usar a seguinte regra para fazer isso:

```
# iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE$
```

Você também precisará habilitar o IP forwarding. Você pode fazer isso temporariamente, usando o seguinte comando:

```
# echo 1> /proc/sys/net/ipv4/ip\_forward
```

Para habilitar o IP forwarding de uma forma permanente (ou seja, para que a mudança seja mantida após o reboot), você precisará abrir o arquivo */etc/rc.d/rc.inet2* em seu editor de textos favorito e mudar a seguinte linha:

```
IPV4\_FORWARD=0
```

...para isso:

```
IPV4\_FORWARD=1
```

Para mais informações sobre NAT, veja o NAT HOWTO.

14.2.2 tcpwrappers

Os *tcpwrappers* controlam o acesso aos daemons de acordo com a aplicação, e não de acordo com o IP. Isso pode trazer uma camada extra de segurança quando controles de acesso por IP (ex. Netfilter) não estão funcionando corretamente. Por exemplo, se você recompilar o kernel mas esquecer de incluir o suporte ao iptables, sua proteção no nível de IP falhará mas os tcpwrappers ainda o ajudarão a proteger o seu sistema.

Os acessos aos serviços protegidos pelos tcpwrappers podem ser controlados usando os arquivos */etc/hosts.allow* e */etc/hosts.deny*.

A maioria das pessoas poderão ter uma única linha em seus arquivos */etc/hosts.deny* para barrar o acesso a todos os daemons como padrão. Essa linha ficaria assim:

```
ALL : ALL
```

Quando isso é feito, você pode se concentrar em permitir o acesso a serviços para máquinas, domínios, ou faixas de IP específicos. Isso pode ser feito no arquivo */etc/hosts.allow*, que segue o mesmo formato.

Muitas pessoas começariam aceitando todas as conexões do localhost. Isso pode ser feito da seguinte forma:

```
ALL : 127.0.0.1
```

Para permitir o acesso ao SSHd a partir dos endereços 192.168.0.0/24, você poderá usar qualquer uma das duas regras a seguir:

```
sshd : 192.168.0.0/24 sshd : 192.168.0.
```

Também é possível restringir o acesso a máquinas em certos domínios. Isso pode ser feito usando a seguinte regra (note que isso se apoia na veracidade da entrada de DNS reverso para a máquina que está conectando, logo eu não recomendaria seu uso em máquinas conectadas a Internet):

```
sshd : .slackware.com
```

14.3 Mantendo-se Atualizado

14.3.1 Lista de emails slackware-security

Sempre que um problema de segurança afeta o Slackware, um email é mandado a todos os assinantes da lista através do endereço *slackware-security@slackware.com*. Relatórios são enviados sobre vulnerabilidades em qualquer parte do Slackware, menos as relativas aos pacotes contidos no */extra* ou no */pasture*. Esses emails de notificação de segurança incluem detalhes sobre como obter as versões atualizadas dos pacotes do Slackware ou work-arounds, se existirem.

Informações sobre como se inscrever nas listas de email do Slackware podem ser encontradas na Seção 2.7.

14.3.2 O diretório /patches

Sempre que são liberados pacotes atualizados para uma versão do Slackware (normalmente para corrigir um problema de segurança, no caso de versões do Slackware já liberadas), eles são colocados no diretório */patches*. O caminho completo para esses patches vai depender do mirror que você estiver utilizando, mas terão a seguinte forma */caminho/para/slackware-x.x/patches/*.

Antes de instalar esses pacotes, é uma boa idéia verificar o *md5sum* do pacote. O *md5sum*(1) é um utilitário de linha de comando que cria uma "chave"matemática "única"para o arquivo. Se um único bit do arquivo tiver sido modificado, ele gerará um valor diferente de md5sum para o arquivo.

```
$ md5sum package-<ver>-<arch>-<rev>.tgz
6341417aa1c025448b53073a1f1d287d   package-<ver>-<arch>-<rev>.tgz
```

Você deverá comparar esse valor com o valor contido no arquivo *CHECKSUMS.md5* na raiz do diretório *slackware-$VERSION* (também contido no diretório */patches* para os patches) ou no email da lista *slackware-security* para o pacote em questão.

Se você tiver um arquivo com os valores md5sum nele, você pode usa-lo como fonte dos md5sum com a opção *-c* do comando *md5sum*.

```
# md5sum -c CHECKSUMS.md5
./ANNOUNCE.10_0: OK
./BOOTING.TXT: OK
./COPYING: OK
./COPYRIGHT.TXT: OK
./CRYPTO_NOTICE.TXT: OK
./ChangeLog.txt: OK
./FAQ.TXT: FAILED
```

Como você pode ver, qualquer arquivo com o *md5sum* correto é listado como *OK* enquanto os arquivos que estão com o md5sum incorreto são listados como "*FAILED*". (Sim, isso foi um insulto a sua inteligência. Porque você mexeu comigo?)

Capítulo 15

Arquivos Agrupados e Compactados

15.1 gzip

O *gzip*(1) é o programa de compressão da GNU. Ele recebe um arquivo e então o comprime. Seu uso básico é como a seguir:

```
$  gzip nome_do_arquivo
```

O nome do arquivo resultante será *nome_do_arquivo*.gz// e ele geralmente será menor que o arquivo de entrada. Observe que *nome_do_arquivo* será substituído por *nome_do_arquivo.gz*. Isto significa que *nome_do_arquivo* não existirá mais, somente existirá sua cópia comprimida. Arquivos texto regulares serão bem compactados, enquanto que imagens jpeg, arquivos mp3 e outros arquivos como estes não serão tão compactados, pois eles já estão em um formato compactado. Esta forma básica de usar o programa de compressão mantém um balanceamento entre o tamanho final do arquivo e o tempo de compressão. A compressão máxima pode ser obtida da seguinte maneira:

```
$  gzip -9 nome_do_arquivo}
```

Desta maneira a compressão do arquivo será mais lenta, mas o arquivo resultante será tão pequeno quanto o*gzip* é capaz de comprimí-lo. Usando valores mais baixos na opção do comando, a compressão será mais rápida, mas o arquivo não será tão compactado.

Para descomprimir arquivos compactados com o *gzip* podem ser usados dois comandos, que são na verdade um mesmo programa. *gzip* irá descomprimir qualquer arquivo com um extensão reconhecida. As extensões reconhecidas são: *.gz*, *-gz*, *.z*, *-z*, *.Z*, ou *-Z*. A primeira maneira é executar *gunzip*(1) sobre um arquivo, da seguinte forma:

```
$  gunzip nome_do_arquivo.gz
```

Isto irá deixar, no diretório corrente, uma versão descomprimida do arquivo de entrada, e a extensão *.gz* será retirada do nome do arquivo. *gunzip* é de fato parte do *gzip* e é idêntico a *gzip -d*. Dessa forma, *gzip* é geralmente pronunciado *gunzip*, uma vez que o nome soa mais legal assim. :ˆ)

15.2 bzip2

O *bzip2*(1) é um programa de compressão alternativo instalado no Slackware Linux. Ele utiliza um algoritmo de compressão diferente do utilizado em *gzip*, o que resulta em algumas vantagens e algumas desvantagens. A principal vantagem em favor de *bzip2* é o tamanho do arquivo compactado. *bzip2* quase sempre irá comprimir melhor que *gzip*. Em alguns casos, isto pode resultar em arquivos dramaticamente menores. Isto pode ser uma grande vantagem para pessoas com conexões lentas. Também lembre-se que, ao copiar um programa de um servidor

FTP público, é, em geral, uma boa regra de etiqueta na Internet pegar arquivos *.bz2* em vez de arquivos *.gz*, já que isto resulta em uma menor sobrecarga, ajudando assim as generosas pessoas que hospedam o servidor.

A desvantagem de *bzip2* é que ele utiliza mais recursos de processamento do que o *gzip*. Isto significa que descomprimir um arquivo com *bzip2* geralmente será mais demorado e utilizará mais o processador do que o *gzip* o faria. Na escolha de um programa de compressão a ser usado, você precisa considerar o tempo de compressão e o tamanho do arquivo compactado e determinar o que é mais importante.

O uso de *bzip2* é quase idêntico ao uso de *gzip*, de maneira que não será gasto muito tempo discutindo isso. Como *gunzip*, *bunzip2* é idêntico a *bzip2 -d*. A principal diferença na prática é que *bzip2* usa a extensão *.bz2*.

```
$  bzip2 nome_do_arquivo
$  bunzip2 nome_do_arquivo.bz2
$  bzip2 -9 nome_do_arquivo
```

15.3 tar

tar(1) é o arquivador em fita da GNU (as fitas já foram a principal mídia para armazenar grandes volumes de dados!). Neste contexto, arquivar significar agrupar e comprimir um conjunto de arquivos para transferência ou armazenamento. O *tar* recebe como entrada vários arquivos ou diretórios e cria um único grande arquivo, também chamado de *pacote*. Isto lhe permite comprimir uma árvore de diretório inteira, o que é impossível apenas utilizando *gzip* ou *bzip2*. O *tar* tem muitas opções de linha de comando, que são explicadas na *página de manual* correspondente. Esta seção irá abordar apenas os casos mais comuns de uso do *tar*.

O uso mais comum para o *tar* é extrair (descomprimir e desagrupar) um pacote que você copiou de um servidor WEB ou FTP. A maioria desses pacotes terão a extensão *.tar.gz*. Um pacote é comumente conhecido como um *tarball*. Isto significa que vários arquivos foram agrupados utilizando-se o *tar* e então compactados com o *gzip*. Você também poderá encontrar tais pacotes com a extensão *.tar.Z*. Esta extensão significa a mesma coisa, mas é geralmente encontrada em sistemas Unix mais antigos. Por outro lado, você pode encontrar um pacote com a extensão *.tar.bz2* em algum lugar. O código fonte do *kernel* (núcleo do sistema operacional) é distribuído desta forma, pois o pacote resultante é menor. Como você pode ter imaginado, esse pacote resulta de vários arquivos compactados com o *bzip2* e agrupados com o *tar*. Você pode obter todos os arquivos neste pacote utilizando o *tar* com alguns parâmetros. Para se *desempacotar* um *tarball* utiliza-se a opção *-z* como parâmetro, o que significa que o *gunzip* será utilizado para descomprimir os arquivos. A maneira mais comum para descomprimir e separar os arquivos de tais pacotes é a seguinte:

```
$  tar -xvzf nome_do_arquivo.tar.gz
```

São absolutamente poucas opções. Então o que elas significam? A opção *-x* significa extrair. Ela é importante, uma vez que diz exatamente para o *tar* o que fazer com o arquivo de entrada. Neste caso, nós dividiremos o pacote separando de volta os arquivos que foram agrupados. O *-v* significa ser detalhado, ao invés de ser resumido. Dessa forma, serão listados todos os arquivos que estão sendo desagrupados. É perfeitamente aceitável não usar esta opção se ela for algo que lhe incomoda. De outra forma, você poderia usar *-vv* para a execução ser muito mais detalhada e listar ainda mais informações sobre cada arquivo sendo extraído. A opção *-z* diz ao *tar* para primeiro descomprimir o pacote *nome_do_arquivo.tar.gz* com o *gunzip*. E finalmente, a opção *-f* indica ao *tar* que a próxima seqüencia de caracteres na linha de comando representa o nome do arquivo sobre o qual ele executará as operações.

Existem algumas outras maneiras de se escrever este comando. Em sistemas mais antigos carentes de uma cópia decente do GNU *tar*, você pode encontrá-lo expresso assim:

```
$  gunzip nome_do_arquivo.tar.gz | tar -xvf -
```

Este comando irá descomprimir o arquivo e enviar a saída para o *tar*. Uma vez que o *gzip* escreverá sua saída na saída padrão se assim lhe for inidicado, este comando escreverá o arquivo descompactado na saída padrão. O redirecionamento (|) então envia esta saída para o *tar* para desagrupamento. O - indica para o comando operar sobre a entrada padrão. Ele irá desagrupar o fluxo de dados que receber do *gzip*, escrevendo a saída no disco.

Outra maneira de escrever o primeiro comando é retirando-se o traço que vem antes das opções, como a seguir:

```
$ tar xvzf nome_do_arquivo.tar.gz
```

Você também pode encontrar um pacote compactado com *bzip2*. A versão do *tar* que vem com o Slackware Linux pode manipular esses pacotes assim como os compridos com *gzip*. Ao contrário da opção *-z*, você deve usar *-j*:

```
$  tar -xvjf nome_do_arquivo.tar.bz2
```

É importante notar que o *tar* colocará os arquivos extraídos no diretório corrente. Então, se você tem um pacote em */tmp* que você quer extrair no seu diretório pessoal, existem poucas opções. Primeiro, poderia-se mover o pacote para o seu diretório pessoal e então executar o *tar* sobre ele. Segundo, você poderia especificar o caminho para o pacote na linha de comando. Terceiro, você pode usar a opção *-C* para *explodir* o pacote em um determinado diretório.

```
$  cd $HOME
$  cp /tmp/nome_do_arquivo.tar.gz .
$  tar -xvzf nome_do_arquivo.tar.gz
$  cd $HOME
$  tar -xvzf /tmp/nome_do_arquivo.tar.gz
$  cd /
$  tar -xvzf /tmp/nome_do_arquivo.tar.gz -C $HOME
```

Todos os comandos acima são equivalentes. Em cada caso, os arquivos do pacote são extraídos em seu diretório pessoal e o pacote original compactado permanecerá em seu lugar inicial.

Sendo assim, o que adiante ser capaz de extrair estes pacotes se você não é capaz de criar um deles? Bem, o comando *tar* também cuida disso. Na maioria dos casos isto é tão fácil quanto remover a opção *-x* e substituí-la pela opção *-c*.

```
$  tar -cvzf nome_do_arquivo.tar.gz .
```

Neste comando, a opção *-c* diz ao *tar* para agrupar os arquivos criando um pacote, enquanto que a opção *-z* executa o *gzip* sobre o pacote para comprimí-lo. *nome_do_arquivo.tar.gz* será o nome do pacote que você quer criar.

Não é sempre necessário especificar a opção *-f*, mas de qualquer maneira é uma boa prática fazê-lo. Sem isso, o comando *tar* envia sua saída para a saída padrão, o que é geralmente desejável para redirecionamento da saída do *tar* para outro programa, como a seguir.

```
$  tar -cv . | gpg --encrypt
```

Este comando cria um pacote não compactado que agrupa o conteúdo do diretório corrente e então redireciona esse pacote para o *gpg* que criptografa e comprime o pacote, fazendo com que seja praticamente impossível sua leitura por outra pessoa além daquela que conhece a chave secreta.

15.4 zip

Por fim, existem dois utilitários que podem ser usados em arquivos *zip*. Estes arquivos são muito comuns no mundo Windows, de maneira que o Linux tem programas para lidar com eles. O arquivo de compressão é denominado *zip*(1), e o de descompressão é denominado *unzip*(1).

```
$  zip teste *
```

Isto criará o arquivo *teste.zip*, que irá conter todos os arquivos do diretório corrente. O *zip* adicionará a extensão *.zip* automaticamente, de forma que não há necessidade de incluí-la no nome do arquivo. Você também pode recursivamente comprimir os subdiretórios existentes:

```
$  zip -r teste *
```

Bem como, descomprimir arquivos também é fácil.

```
$  unzip teste.zip
```

Isto irá extrair todos os arquivos e diretórios no arquivo *teste.zip*.

Os utilitários *zip* têm várias opções avançadas para criar pacotes que podem se extrair automaticamente, excluir arquivos, controlar o tamanho do arquivo compactado, mostrar o que acontecerá e muito mais. Veja as *páginas de manual* dos comandos *zip* e *unzip* para descobrir como usar essas opções.

Capítulo 16

Vi

O *vi*(1) é o programa editor de texto padrão do Unix, e apesar de que dominá-lo não seja tão essencial quanto já foi, é ainda algo que vale a pena. Há algumas versões (ou clones) do *vi* disponíveis, incluindo *vi*, *elvis*,*vile*, e o *vim*.

Ao menos um destes está disponível em qualquer versão do Unix, assim como no Linux. Todas estas versões incluem os mesmos recursos e comandos básicos, logo, aprender a usar um clone deve facilitar o aprendizado de outro. Com a variedade de editores de texto inclusos nas distribuições do Linux e variantes do Unix atualmente, muitas pessoas não utilizam mais o *vi*. De qualquer maneira, ele permanece sendo o editor de texto mais universal do Unix e sistemas derivados. Dominar o *vi* significa que você nunca mais se sentará diante de uma máquina Unix e não se sentirá confortável com pelo menos um poderoso editor de texto.

O *vi* inclui diversos recursos como o destaque de sintaxe, formatação de código, um poderoso mecanismo de busca e substiuição, macros e outros. Estes recursos o fazem especialmente atrativo para programadores, desenvolvedores web e afins. Administradores de sistema vão apreciar a possível automação e integração com o shell.

No Slackware Linux, a versão padrão do *vi* disponível é o *elvis*. Outras versões - incluindo o *vim* e o *gvim* - estão disponíveis se você tiver os pacotes apropriados instalados. O *gvim* é uma versão gráfica do *vim* que inclui barras de tarefas, menus destacáveis e caixas de diálogo.

16.1 Iniciando o vi

O *vi* pode ser iniciado a partir da linha de comando de várias maneiras. A mais simples dela é apenas:

```
$ vi
```

Uma sessão do vi.

```
 */
void
TabReset(Tabs tabs)
{
    int i;

    for (i = 0; i < TAB_ARRAY_SIZE; ++i)
        tabs[i] = 0;

    for (i = 0; i < MAX_TABS; i += 8)
        TabSet(tabs, i);
}
tabs.c
    if ((tmp = TypeCallocN(Char, length)) == 0)
        SysError(ERROR_SREALLOC);
    *sbufaddr = tmp;

    for (i = k = 0; i < nrow; i++) {
        k += BUF_HEAD;
        for (j = BUF_HEAD; j < old_max_ptrs; j++) {
            memcpy(tmp, base[k++], ncol);
            tmp += ncol;
        }
screen.c
```

Isto irá iniciar o *vi* com um buffer vazio. Neste ponto, você verá uma tela com quase nada escrito. O vi está agora no *modo de comando*, esperando alguma ação sua. Para uma discussão dos diversos modos do *vi*, veja a seção 16.2. Para finalizar a sessão e sair do *vi*, digite o seguinte:

```
:q
```

Assumindo que não tenha havido modificações no arquivo, este comando irá fechar o *vi*. Se houve modificações, este comando irá adverti-lo e lhe dizer como ignorá-las. Ignorar modificações usualmente significa anexar um ponto de exclamação após o ***q*** resultando em:

```
:q!
```

O ponto de exclamação usualmente significa forçar alguma ação. Nós discutiremos esta e outras combinações mais adiante.

Você também pode inciar o *vi* com um arquivo pré-existente. Por exemplo, o arquivo */etc/resolv.conf* poderia ser aberto da seguinte maneira:

```
$ vi /etc/resolv.conf
```

E finalmente, o *vi* pode ser iniciado em uma linha particular de um arquivo. Isto é especialmente útil para programadores quando uma mensagem de erro inclui a linha na qual houve o erro. Por exemplo, você poderia iniciar o *vi* na linha 47 do arquivo */usr/src/linux/init/main.c* da seguinte maneira:

```
$ vi +47 /usr/src/linux/init/main.c
```

O *vi* mostrará o arquivo especificado e irá colocar o cursor na linha especificada. No caso em que você especificar uma linha que esteja após o final do arquivo, o *vi* colocará o cursor na última linha. Isto é especialmente útil para programadores, pois eles podem pular exatamente para o local no arquivo no qual ocorreu um erro, sem ter que procurar por ele.

16.2 Modos

O *vi* opera em vários modos, que são usados para cumprir várias tarefas. Quando você inicia o *vi*, você é colocado dentro do modo de comando. Deste ponto, você pode lançar diversos comandos para manipular texto, mover-se pelo arquivo, salvar, sair, e alternar entre os outros modos. A edição de texto é feita em modo de inserção. Você pode rapidamente alternar entre os modos com uma variedade de combinações de teclas, que são explicadas abaixo.

16.2.1 Modo de Comando

Ao entrar no vi você é primeiramente colocado no modo de comando. A partir deste modo, você não pode digitar texto ou editar algo que já esteja na tela diretamente. Entretanto, você pode manipular o texto, pesquisar, fechar, salvar, carregar novos arquivos e muito mais. O texto que segue será apenas uma introdução ao modo de comando. Para uma descrição dos vários comandos, veja seção 16.7.

Provavelmente o comando mais usado no modo de comando é a alternação para o modo de inserção. Isto é feito pressionando-se a tecla **i**. O cursor muda de formato, e -- INSERT –"é mostrado no inferior da tela (note que isto não acontece em todos os clones do *vi*). De lá, todas os caracteres digitados são enviados ao buffer corrente e são mostrados na tela. Para voltar ao modo de comando, pressione a tecla **ESCAPE**.

O modo de comando é também usado ao mover-se pelo arquivo. Em alguns sistemas, você pode usar as setas do teclado para mover-se. Em outros sistemas, você pode precisar usar as teclas ***hjkl***. Aqui está uma simples listagem de como as teclas são usadas para mover-se pelo arquivo:

h	move-se um caractere à esquerda
j	move-se para um caractere abaixo
k	move-se para um caractere acima
l	move-se um caractere à direita

Simplesmente pressione uma tecla para mover-se. Como você verá posteriormente, estas teclas podem ser combinadas com um número para mover-se de forma mais eficiente.

Muitos dos comandos que você usará no modo de comando iniciam-se com dois-pontos. Por exemplo, para sair, teclamos **:q**, como explicado anteriormente. Os dois-pontos simplesmente indicam que trata-se de um comando, enquanto que a letra ***q*** diz ao *vi* para encerrar a sessão e sair. Outros comandos são formados por um número opcional, seguido por uma letra. Estes comandos não tem dois-pontos antes deles, e geralmente são usados para manipular o texto.

Por exemplo, para apagarmos uma linha de um arquivos digitamos **dd**. Isto irá remover a linha na qual está o cursor. Lançando o comando **4dd** diria ao *vi* para remover a linha na qual está o cursor e as 3 seguintes. Em geral, o número diz ao *vi* quantas vezes executar o comando dado.

Você pode combinar um número com as teclas de movimento para mover-se por vários caracteres de uma só vez. Por exemplo, **10k** moveria o cursor 10 linhas acima na tela.

O modo de comando também pode ser usado para copiar e colar, inserir texto e ler outros arquivos no buffer corrente. Para copiar um trecho do texto usamos a tecla **y** (**y** vem da palavra inglesa yank, que em portugues significa "arrancar", "extrair"). Para copiarmos a linha corrente

teclamos **yy**, e este comando pode ser prefixado com um número para extrair mais linhas. Então, mova-se para o local para o qual o texto extraido deve ser copiado e pressione a tecla **p**. O texto é colado na linha posterior à linha corrente.

Para cortar uma parte do texto digitamos **dd**, e a tecla **p** pode ser usada para colar o texto recortado novamente no arquivo. Ler texto de outro arquivo é um procedimento simples. Apenas digite **:r**, seguido por um espaço e o nome do arquivo que contém o texto a ser inserido. O conteúdo do arquivo será colado no buffer corrente na linha após a que se encontra o cursor. Os clones mais sofisticados do *vi* também contém complementação de nome de arquivo similiar à do shell.

O uso final que será mostrado é a pesquisa. O modo de comando permite pesquisas simples, assim como comandos complicados de busca e substituição que usufruem de uma poderosa versão de expressões regulares. Uma completa discussão sobre expressões regulares está além do escopo deste capítulo, logo, esta seção cobrirá somente simples meios de pesquisa.

Uma simples busca é feita pressionando-se a tecla /, seguida do texto pelo qual você está procurando. O *vi* irá fazer a pesquisa do ponto em que está o cursor até o fim do arquivo, parando no momento em que a pesquisa for satisfeita. Note que buscas inexatas irão fazer com que o *vi* pare. Por exemplo, uma busca por *"the"* irá fazer com que o *vi* pare em *"then"*, *"therefore"*, e em outras palavras. Isto acontece porque todas estas palavras contêm o trecho de pesquisa *"the"*.

Após o *vi* encontrar o primeiro termo que casa com o termo de pesquisa, você pode continuar pesquisando pelo termo simplesmente pressionando a tecla / seguida de um enter. Você também pode pesquisar se o termo de pesquisa existe em algum trecho anterior à localização do cursor substituindo a barra pelo caractere **?**. Por exemplo, para procurar por *"the"* em algum ponto anterior ao cursor, digitamos **?the**.

16.2.2 Modo de inserção

A inserção e substituição de texto é feita no modo de inserção. Como discutido anteriormente, você pode ir para o modo de inserção pressionando a tecla **i** no modo de comando. Então, todo o texto que você digita é enviado ao buffer corrente. Pressionar a tecla **ESCAPE** lhe levará ao modo de comando.

A substituição de texto pode ser feita de algumas maneiras. Da linha de comando, pressionando-se a tecla **r** permitirá que você substitua um caractere que está sob o cursor. Apenas digite o novo caractere e ele substituirá o caractere que está sob o cursor pelo caractere digitado. Você será então imediatamente mandado devolta ao modo de comando. Pressionando-se a tecla **R** você poderá substituir quantos caracteres você quiser. Para sair deste modo de substituição, apenas tecle **ESCAPE** para retornar ao modo de comando.

Ainda há outra maneira de se alternar entre o modo de inserção e o de substituição. Pressionando-se a tecla **INSERT** a partir do modo de comando, você será levado ao modo de inserção. Uma vez que você se encontra no modo de inserção, a tecla **INSERT** servirá como uma tecla de alternância entre o modo de inserção e substituição. Ao pressioná-la pela primeira vez você estará no modo de substituição. Pressionando-a novamente você será enviado ao modo de inserção.

16.3 Abrindo Arquivos

O *vi* permite que você abra arquivos a partir do modo de comando assim como se especifica a abertura de um arquivo pela linha de comando. Para abrir o arquivo */etc/lilo.conf*:

```
:e /etc/lilo.conf
```

Se você tiver feito alterações no buffer corrente que não tenham sido salvas, o *vi* irá reclamar. Você ainda pode abrir o arquivo sem salvar o buffer corrente digitando **:e!**, seguido de um espaço

e do nome do arquivo. Em geral, as advertências do *vi* podem ser suprimidas adicionando-se um ponto de exclamação no final do comando.

Se você quiser reabrir o arquivo corrente, você pode fazê-lo simplesmente digitando **e!**. Isto é particularmente útil caso você tenha bagunçado o arquivo de alguma maneira e queira reabri-lo.

Alguns clones do *vi* (por exemplo, o *vim*) permitem que múltiplos buffers sejam abertos ao mesmo tempo. Por exemplo, para abrir o arquivo *09-vi.sgml* que está no meu diretório home enquanto outro arquivo está aberto, eu poderia digitar:

```
:split \~{}/09-vi.sgml
```

O novo arquivo é exibido na metade superior da tela, e o arquivo que já estava aberto é exibido na metade inferior da tela. Há vários comandos para manipulação da tela dividida, e muitos destes comandos lembram o *Emacs* O melhor lugar para procurar por estes comandos seria a página de manual do seu clone do vi. Note que muitos clones não suportam a idéia de divisão de tela, então você pode não estar apto a utilizá-la.

16.4 Salvando Arquivos

Há algumas maneiras de se salvar arquivos no *vi.* Se você quiser salvar o buffer corrente para o arquivo *bah*, você poderia digitar:

```
:w bah
```

Uma vez que você tenha salvado o arquivo pela primeira vez, para salvá-lo novamente basta digitar **:w**. Qualquer alteração será gravada no arquivo. Após você ter salvado o arquivo, você é jogado de volta para dentro do modo de comando. Se você quiser salvar o arquivo e encerrar o *vi* (uma operação muito comum), você podeira digitar **:wq**. Que diz ao *vi* para salvar o buffer corrente para o arquivo e encerrar a sessão, voltando ao shell.

Se por acaso, você quiser salvar um arquivo cujas permissões sejam apenas de leitura, você poderia fazê-lo adicionando um ponto de exclamação após o comando de gravação, da seguinte forma:

```
:w!
```

Entretanto, haverá casos em que você não poderá gravar no arquivo (por exemplo, caso você esteja tentando editar um arquivo que seja de outro usuário). Quando isto acontecer, o *vi* irá lhe dizer que não foi possível salvar o arquivo. Se você realmente quiser editar o arquivo, você terá que editá-lo como usuário *root* ou (de preferência) como o usuário dono do arquivo.

16.5 Encerrando o vi

Uma maneira de encerrar o *vi* é através do comando **:wq**, que irá salvar o buffer corrente para o arquivo antes de encerrar a sessão. Você também pode encerrar a sessão sem salvar as alterações feitas no arquivo com o comando **:q** ou (mais comum) com o comando **:q!**. A segunda maneira é utilizada quando você tiver modificado o arquivo, mas não desejar salvar as alterações feitas.

Pode acontecer da sua máquina ou o *vi* travarem. Entretanto, ambos *elvis* e o *vim* tentam minimizar os danos. Ambos os editores salvam ocasionalmente o buffer aberto para um aquivo temporário. Este arquivo é usualmente nomeado de forma similar ao arquivo aberto, mas com um ponto no início do nome. Isto faz com que o arquivo seja oculto.

Este arquivo temporário é removido uma vez que o editor é encerrado sob condições normais. Isto significa que a cópia temporária existirá para o caso de haver algum problema. Quando você voltar a editar o arquivo novamente, você será questionado sobre qual ação tomar. Na maioria dos casos, uma larga quantia do seu trabalho não salvo pode ser recuperada. O *elvis* também lhe enviará um e-mail informando da existência de uma cópia de backup.

16.6 Configuração do vi

O seu clone do *vi* pode ser configurado de várias maneiras.

Uma variedade de comandos podem ser dados no modo de comando para configurar o *vi* do jeito que você gosta. Dependendo do seu editor, você pode habilitar recursos para facilitar a programação (como o destaque de sintaxe, auto-identação, e outros mais), configurar macros para automatizar tarefas, habilitar substituições textuais, e muito mais.

Quase todos estes comandos podem ser colocados dentro de um arquivo de configuração no seu diretório home. No *elvis* seria o arquivo *.exrc*, enquanto que no *vim* seria o arquivo *.vimrc*. A maioria dos comandos de configuração que podem ser digitados no modo de comando, podem ser colocados no arquivo de configuração. Isto inclui informações de configuração, substituições textuais, macros, e mais.

A discussão de todas essas opções e as diferenças entre os editores é um assunto muito complicado. Para maiores informações, verifique a página de manual ou o site do seu editor *vi* predileto. Alguns editores (como o *vim*) possuem uma extensa documentação dentro do editor que pode ser acessada com o comando **:help**, ou algo similar. Você também pode checar o livro da O'Reilly chamado *Learning the //vi* Editor// escrito por Lamb e Robbins.

Muitos programas comuns do Linux carregarão um arquivo texto no *vi* por padrão. Por exemplo, ao se editar as tarefas agendadas do sistema (crontabs) será iniciado o *vi* por padrão. Se você não gosta do *vi* e gostaria que outro editor fosse iniciado ao invés dele, tudo o que você precisa fazer é setar a variável de ambiente *VISUAL* para o editor de sua preferência. Para maiores informações sobre como setar variáveis de ambiente, veja a seção chamada Variáveis de Ambiente no capítulo 8. Se você deseja ter certeza de que o seu editor será o padrão toda vez que você logar-se, adicione sete a variável VISUAL dentro dos seus arquivos *.bash_profile* ou *.bashrc*.

16.7 Comandos do Vi

Esta seção é uma referência rápida aos comandos mais comuns do *vi*. Alguns destes já foram discutidos anteriormente no capítulo, enquanto outros não.

Movimento

Operação	**Tecla**
esquerda, abaixo, acima, direita	**h**, **j**, **k**, **l**
Para o fim da linha	**$**
Para o início da linha	ˆ
Para o fim do arquivo	**G**
Para o início do arquivo	**:1**
Para a linha 47	**:47**

Editando

Operação	**Tecla**
Removendo uma linha	**dd**
Removendo cinco linhas	**5dd**
Substituindo um caractere	**r**
Removendo um caractere	**x**
Removendo dez caracteres	**10x**
Defazer a última ação	**u**
Unir linhas atual e seguintes	**J**
Substituir velho por novo, globalmente	**%s'velho'novo'g**

Localizando

Operação	**Tecla**
Procurar por *asdf*	**/asdf**
Procurar atrás por *asdf*	**?asdf**
Repetir a última busca para frente	**/**
Repetir a última busca para trás	**?**
Repetir a última busca, na mesma direção	**n**
Repetir a última busca, na direção oposta	**N**

Salvando e Saindo

Operação	**Tecla**
Sair	**:q**
Sair sem salvar	**:q!**
Salvar e sai	**:wq**
Salvar, sem sair	**:w**
Recarregar o arquivo atualmente aberto	**:e!**
Salvar o buffer para um arquivo *asdf*	**:w asdf**
Abrir o arquivo *hejaz*	**:e hejaz**
Ler o arquivo *asdf* dentro do buffer	**:r asdf**
Ler a saída do comando *ls* dentro do buffer	**:r !ls**

,

Capítulo 17

Emacs

Enquanto o *vi* (juntamente com seus clones) é sem dúvida o editor mais presente em sistemas derivados do Unix, O Emacs vem em segundo lugar. Ao invés de usar diferentes *"modos"*, como o *vi*, ele utiliza combinações das teclas **Control** e **Alt** para dar comandos, da mesma forma que você pode usar combinações das teclas **Control** e **Alt** em um outro processador de textos e em muitas outras aplicações afim de executar certas funções. (Apesar de ser notável que os comandos raramente se correspondem; então enquanto muitas aplicações modernas utilizam **Ctrl C** / **X**/ **V** para copiar, recortar e colar, o Emacs utiliza teclas diferentes e atualmente um mecanismo um tanto quanto diferente para isto.)

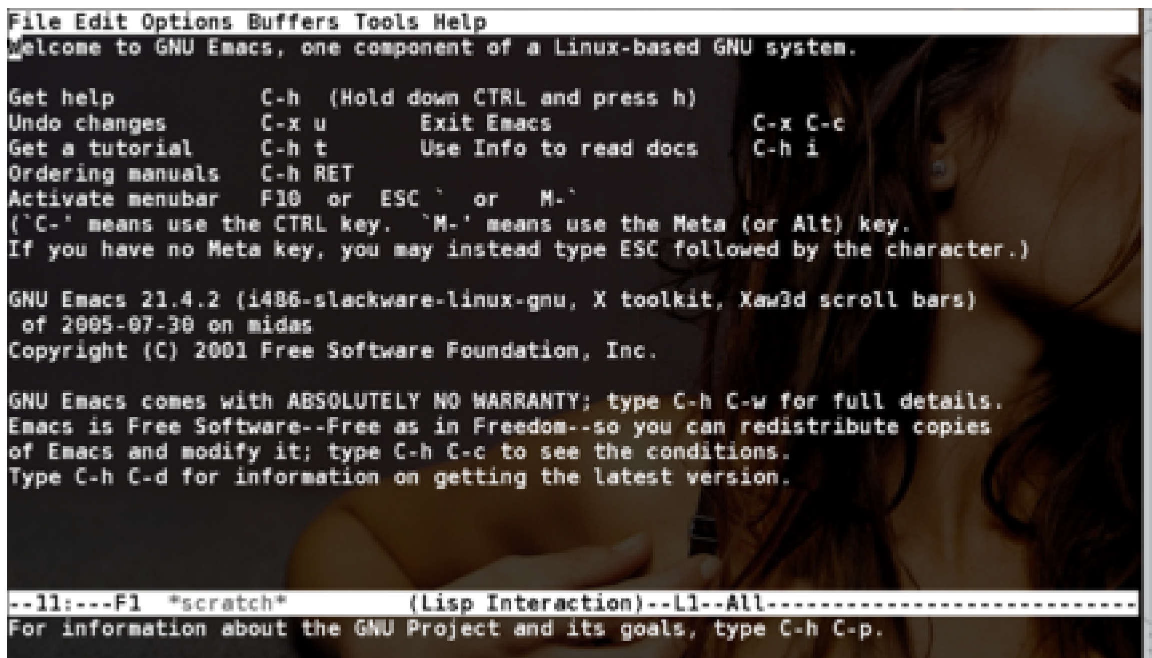
Também diferentemente do *vi*, que é um (excelente) editor e nada mais, o Emacs é um programa com quase infinitas capacidades. O Emacs é (em sua maior parte) escrito em linguagem Lisp, que é uma linguagem de programação muito poderosa que tem a característica peculiar de que todo programa escrito nela é automaticamente um compilador Lisp em si. Isto significa que o usuário pode estender o Emacs, e de fato escrever programas inteiramente novos *no Emacs*.

Como resultado disto, o Emacs não é mais somente um editor. Há muitos pacotes adicionais para o Emacs disponíveis (o código fonte do programa vem com o programa) que provêem todas as espécies de funcionalidades. Muitos destas são relacionados à edição de texto, que é dentre todas as demais, a tarefa básica do Emacs mas que não se limitam a isto. Existem por exemplo alguns programas de planilhas eletrônicas para o Emacs, base de dados, jogos, clientes de e-mail e notícias (sendo o Gnus o mais famoso deles), etc.

Existem duas versões principais do Emacs: O GNU Emacs (que é a versão que vem com o Slackware) e o XEmacs. O último *não* é uma versão do Emacs para rodar sob o X. De fato, ambos Emacs e XEmacs rodam no console assim como sob o X. O XEmacs foi iniciado como um projeto para limpar e arrumar o código fonte do Emacs. Atualmente, ambas as versões estão sendo ativamente desenvolvidas, e há de fato muita interação entre os dois times de desenvolvimento. Para o capítulo presente, é indiferente se você usa o Emacs ou o XEmacs, as diferenças entre eles não são relevantes para o usuário final.

17.1 Iniciado o Emacs

O Emacs pode ser inciado no shell simplesmente digitando-se *emacs*. Se você estiver rodando o X, o Emacs irá (normalmente) surgir com sua própria janela, usualmente com uma barra de menu no topo, onde você pode encontrar as funções mais importantes. Na inicialização, o Emacs irá primeiramente mostrar uma mensagem de boas vindas, e então após alguns segundos ele irá jogá-lo no *scratch* buffer. (veja seção 17.2).



Você também pode iniciar o Emacs com um arquivo existente digitando

```
$ emacs /etc/resolv.conf
```

Isto fará com que o Emacs carregue o arquivo especificado quanto inciar, pulando a mensagem de boas vindas.

17.1.1 Teclas de Comando

Como mencionado acima, o Emacs utiliza combinações das teclas **Control** e **Alt** para comandos. A convenção usual é escrever estes com **C letra** e **M- letra**, respectivamente. Então **C x** significa **Control x**, e **M- x** significa **Alt x**. (A letra **M-** é usada ao invés de A porque originalmente não existia a tecla **Alt** e sim a tecla **Meta**. A tecla **Meta** desapareceu dos teclados dos computadores, e no Emacs a tecla **Alt** tomou então a sua função.)

Muitos comandos do Emacs consistem de sequências e combinações de teclas. Por exemplo, **C x C c** (que é **Control x** seguida de **Control c**) encerra o Emacs, **C x C s**salva o arquivo atual. Mantenha em mente que **C x C b** *não* é o mesmo que **C x b**. A primeira significa **Control x** seguida de **Control b**, enquanto a segunda significa **Control x** seguida apenas de '**b**'.

17.2 Buffers

No Emacs, o conceito de *buffers* é essencial. Todo arquivo que você abre é carregado em seu próprio buffer. Além disto, o Emacs possui alguns buffers especiais, que não contém um arquivo, mas são usados para outras coisas. Tais buffers especiais usualmente têm um nome que começa e termina com um asterisco. Por exemplo, o buffer que o Emacs mostra quando é primeiramente inciado, é o tão famoso *scratch* buffer.

No *scratch* buffer, você pode digitar texto normalmente, mas o texto que é digitado não é salvo quando o Emacs é fechado.

Há um outro buffer especial sobre o qual você precisa saber, e este é o minibuffer. Este buffer consiste de apenas uma linha, e está sempre na tela: ele é a última linha da janela do Emacs, abaixo da barra de status para o buffer em uso. O minibuffer é onde o Emacs mostra mensagens para o usuário, e é também o lugar onde os comandos, que requerem alguma entrada do usuário, são executados. Por exemplo, quando você abre um arquivo, o Emacs irá perguntar pelo nome do arquivo no minibuffer. A Mudança de um buffer para outro pode ser feita com

o comando **C x b**. Isto irá questioná-lo sobre o nome de um buffer (o nome de um buffer é geralmente o nome do arquivo que você está editando nele), é dado uma escolha padrão, que é normalmente o nome do buffer no qual você esteve antes de ter mudado para, ou criado, o buffer atual. Você será mandado para este buffer padrão caso pressione a tecla *Enter*.

Se você deseja ir para outro buffer que não seja o padrão sugerido pelo Emacs, basta que você digite o nome do buffer desejado. Note que você pode usar a tão famosa tecla **Tab** para fazer a completação de nome aqui: digite as primeiras letras do nome do buffer e pressione a tecla **Tab**; O Emacs irá então completar o nome do buffer. A completação com a tecla **Tab** funciona em todas as partes do Emacs em que ela faça sentido.

Você pode obter uma lista dos buffers abertos pressionando-se **C x C b**. Este comando irá usualmente dividir a tela em duas, mostrando o buffer no qual você estava trabalhando na metade superior da tela, e o novo buffer chamado *Buffer List* na metade inferior da tela. Este buffer contém uma lista de todos os buffers, seus tamanhos e modos, e os arquivos, se houver algum que aqueles buffers listados estejam visitando. Você pode sair desta tela dividida digitando **C x 1**.

Nota: Sob o X, a lista de buffers também está disponível no menu Buffer na barra de menu.

17.3 Modos

Todo buffer no emacs tem a si associado, um modo. Este modo é muito diferente da idéia de modos do *vi*: um modo lhe diz em que tipo de buffer você se encontra. Por exemplo, existe o modo texto (text-mode) para arquivos normais de texto, porém existem também modos tais como o modo c (c-mode) para edição de programas escritos em C, modo sh (sh-mode) para edição de shell scripts, modo latex (latex-mode) para edição de arquivos latex, modo mail (mail-mode) para edição de e-mail, notícias, mensagens e etc. Um modo provê customizações e funcionalidades especiais que são úteis para o tipo de arquivo que você está editando. Também é possível para um modo, a redefinição de teclas e comandos. Por exemplo, no modo texto, a tecla Tab simplesmente pula para a próxima tabulação e pára, mas em muitos modos de linguagens de programação, a tecla **Tab** identa a linha atual de acordo com a pronfundidade do bloco no qual a linha se encontra.

Os modos mencionados acima são chamados de modos principais. Cada buffer tem exatamente um modo principal. Adicionalmente, um buffer pode ter um ou mais modos secundários. Um modo secundário provê recursos adicionais que podem ser úteis para certas tarefas de edição. Por exemplo, se você pressiona a tecla **INSERT**, você invoca o modo de subscrição (overwrite-mode), que faz o que você supõe que faça. Há também um modo de auto preenchimento (auto-fill-mode), que é útil em combinação com o modo texto (text-mode) ou com o modo latex (latex-mode): ele fará com que cada linha que você digita seja automaticamente embalada, uma vez que a linha alcance um certo número de caracteres. Sem o modo de auto preenchimento (auto-fill-mode), você teria que digitar **M- q** para preencher um parágrafo. (Que você pode também utilizar para reformatar um parágrafo após ter editado algum texto e ele tenha deixado de ser preenchido.)

17.3.1 Abrindo Arquivos

Para abrir um arquivo no Emacs, digite:

```
C-x C-f
```

O Emacs irá lhe perguntar o nome do arquivo que você deseja abrir, preechendo algum caminho padrão para você (que é usualmente ~/). Após você digitar o nome do arquivo que deseja abrir (você também pode usar aqui a tecla **Tab** para completar) e pressionar a tecla **ENTER** , o Emacs irá abrir o arquivo em um novo buffer e mostrar este buffer na tela.

Nota: O Emacs irá automaticamente criar um novo buffer, ele não irá carregar o arquivo no buffer atual.

Para criar um novo arquivo no Emacs, você não pode apenas ir digitando algo. Você primeiramente deve criar um buffer para ele, e dar um nome para o arquivo. Você faz isso digitando **C x C f** e digitando um nome para o arquivo, exatamente como se você estivesse abrindo um arquivo existente. O Emacs irá notar que o arquivo cujo nome você digitou não existe, e criará um novo buffer e informar a criação de um novo arquivo *(New file)* no minibuffer.

Quando você digita **C x C f** e então entra com um nome de um diretório ao invés de um nome de arquivo, o Emacs irá criar um novo buffer no qual você irá encontrar uma lista de todos os arquivos existentes naquele diretório. Você pode mover o cursor para o arquivo que você está procurando e digitar, e o Emacs irá abrí-lo. (Há de fato muitas outras ações que pode tomar aqui, tais como apagar, renomear e mover arquivos, etc. O Emacs está agora no dired-mode, que é basicamente um simples gerenciador de arquivos.)

Quando você estiver digitado **C-x C-f**e derepente mudar de idéia, você pode digitar **C-g** para cancelar a ação. **C-g** funciona em quase todo lugar que você desejar cancelar uma ação ou comando que você estiver iniciado mas que não queira terminar.

17.4 Edição Básica

Quando você tiver aberto um arquivo, você obviamente poderá mover-se ele com o cursor. As **teclas do cursor** e **PgUp**, **PgDn** fazem o que você possa supor. **Home** e **End** pulam para o início e final da linha. (Em versões mais antigas, elas podem pular para o início e fim do buffer.) Entretanto, há também a combinação das teclas **Control** e **Meta** (**Alt**) para mover o cursor. Porque você não precisa mover suas mãos para outra parte do teclado para utilizá-las, elas são muito mais facilmente acessíveis uma vez que você se acostume com elas. Os mais importantes comandos semelhantes estão listados na tabela abaixo.

Comandos Básicos de Edição

Comando*	Resultado**
C-b	volta um caractere
C-f	avança um caractere
C-n	vai para a linha abaixo
C-p	vai para a linha acima
C-a	vai para o início da linha
C-e	vai para o fim da linha
M-b	vai para a palavra anterior
M-f	vai para a palavra seguinte
M-}	vai para o parágrafo seguinte
M-{	vai para o parágrafo anterior
M-a	vai para a sentença anterior
M-e	vai para a sentença seguinte
C-d	apaga o caractere sob o cursor
M-d	apagar até o fim da palavra atual
C-v	abaixa uma tela (assim como a tecla., PgDn)
M-v	sobe uma tela (assim como a tecla., PgUp)
M-<	vai para o início do buffer
M->	vai para o fim do buffer
C-_	desfaz a última alteração (pode ser repetido); note que você realmente tem que digitar **Shi**
C-k	apaga até o fim da linha atual
C-s	efetua uma busca à frente do cursor até o fim do arquivo
C-r	efetua uma busca anteriormente ao cursor até o início do arquivo

Note que muitos comandos com a tecla **Meta** são paralelos aos comandos com a tecla **Control** exceto pelo fato de que eles operam em amplas unidades: enquanto **C-f** avança um caractere, **M-f** avança em uma palavra inteira, etc.

Também note que **M-**<** e **M->** requerem que você digite **Shift Alt comma** e **Shift Alt dot** respectivamente, desde que **<** e **>** estejam ligadas **Shift comma** e **Shift dot**.(A menos é claro que você tenha um teclado em condições diferentes (layout), do teclado padrão US.)

Note que **C k** apagar (apaga, como é freqüentemente chamada) todo o texto após o cursor até o fim da linha, porém não apaga a linha em si (por exemplo, não apaga a indicação de nova linha). Ele apenas apaga a linha se não houver texto após o cursor. Em outras palavras, para apaga uma linha completamente, você tem que posicionar o cursor no início e pressionar **C k** duas vezes: uma para apaga o texto na linha e uma para apagar a linha em si.

17.5 Salvando Arquivos

Para salvar um arquivo, você deve digitar

```
C-x C-s
```

O Emacs não irá lhe solicitar um nome para o arquivo, o buffer atual apenas será salvo para o arquivo que foi carregado inicialmente nele. Se você deseja salvar o seu texto em outro arquivo, digite

```
C-x C-w
```

Quando você salva o arquivo pela primeira vez nesta sessão, o Emacs irá normalmente salvar a antiga versão do seu arquivo em uma cópia de segurança (arquivo de backup), que possui o mesmo nome acompanhado de um sinal de til (˜): então se você está editando um arquivo *//cars.txt//*, o Emacs criará uma cópia de segurança chamada *//cars.txt˜//*. Esta cópia de segurança é uma cópia do arquivo que você abriu. Enquanto você estiver trabalhando, o Emacs irá regularmente criar automaticamente uma cópia do trabalho que você está fazendo, para um arquivo nomeado com sinais de hash: *#cars.txt#*. Esta cópia de segurança é apagada quando você salva o arquivo com C-x C-s. Quando você tesiver finalizado a edição de um arquivo, você poderá matar o buffer que contém o arquivo digitado

```
C-x k
```

O Emacs irá perguntar a você qual buffer você deseja matar, com o buffer atual como padrão, o qual você pode selecionar simplesmente pressionando a tecla **ENTER**. Se você não estiver salvado seu arquivo ainda, o Emacs irá lhe perguntar se realmente deseja matar o buffer.

17.5.1 Saindo do Emacs

Quando você estiver terminado todo o trabalho com o Emacs, você poderá digitar

```
C-x C-c
```

Este comando encerra o Emacs. Se você estiver arquivos não salvos, o Emacs irá lhe dizer isto, e pedir para que você os salve, um por vez. Se você responder não para algum destes, o Emacs irá lhe solicitar uma confirmação final e então sair.

Capítulo 18

Gerenciamento de Pacotes do Slackware

Um pacote de software é um conjunto de arquivos que estão prontos para que você possa instalar. Quando você faz o download de um código fonte, você precisa configurar, compilar, e instalar tudo manualmente. Um pacote de software já está pronto para você utilizar. Tudo o que você precisa fazer é instalar o pacote. Outras características do uso de pacotes de softwares são a facilidade na remoção e atualização do pacote, se você assim o desejar. Slackware possui programas para você gerenciar seus pacotes. Você pode instalar, remover, atualizar, criar, e examinar o conteúdo de um pacote muito facilmente.

Existe um mito sobre o gerenciador de Pacotes da Red Hat, e que o slackware não possui nenhuma ferramenta de gerenciamento de Pacotes. Isto simplesmente não é verdade, pois o Slackware possui uma ferramenta de gerenciamento de pacotes antes mesmo do RedHat existir. Não possui tantas característics como um rpm (ou como um pacote deb), o *pkgtool* e seus programas derivados são tão bons quanto os instaladores de pacotes rpm. A verdade é que o *pkgtool* não faz nenhuma verificação de dependência.

Aparentemente muitas pessoas da comunidade Linux acham que a definição do gerenciamento de pacotes é a de verificar dependências. Isto é muito simples e não é o caso, no Slackware certamente você não terá isso. Isso não significa que os pacotes do Slackware não possuem dependências, mas o gerenciador de pacotes não verifica se existe uma dependência. Gerenciamento de dependência é responsabilidade do Administrador de Sistemas, e isso não cabe a nós resolver.

18.1 Visão Geral do Formato de um Pacote

Antes de aprender os utilitários de pacotes do Slackware, você precisa se familiarizar com o formato do pacote do Slackware. No Slackware, um pacote é simples como um arquivo tar compactado com *gzip*. Os pacotes são feitos para serem extraídos no diretório raiz.

Veja um programa fictício como um exemplo de um pacote do Slackware:

```
./
usr/
usr/bin/
usr/bin/makehejaz
usr/doc/
usr/doc/makehejaz-1.0/
usr/doc/makehejaz-1.0/COPYING
usr/doc/makehejaz-1.0/README
usr/man/
usr/man/man1
```

```
usr/man/man1/makehejaz.1.gz
install/
install/doinst.sh
```

O pacote será extraído no diretório raíz na instalação. Um registro do pacote é adicionado no banco de dados de pacotes onde conterá informações sobre o pacote e será utilizado para atualizar ou remover o pacote posteriormente.

Informações sobre o subdiretório *install/*. Este é um diretório especial que contém um script de pós-instalação chamado *doinst.sh*. Se o sistema encontrar esse arquivo no pacote, ele será executado após a instalação do pacote.

Outros scripts podem ser adicionados no pacote mas serão discutidos com maiores detalhes logo em seguida.

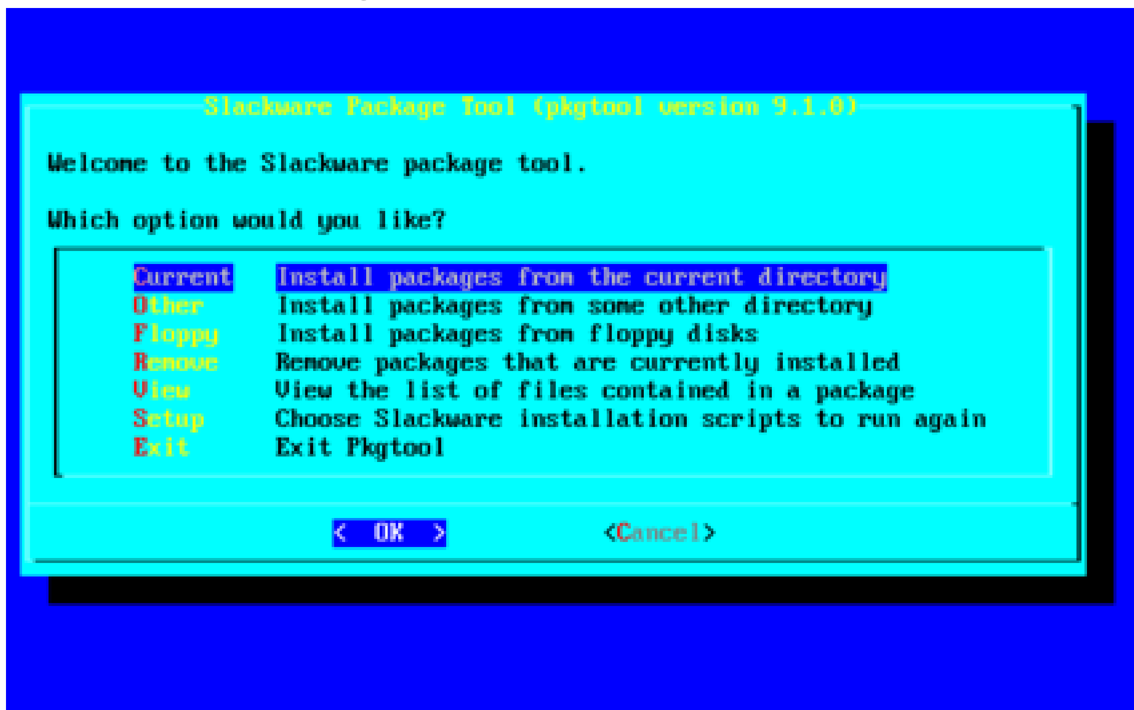
18.2 Utilitários de Pacotes

Existem quatro utilitários principais para o gerenciamento de pacotes. Eles são responsáveis pela instalação, remoção e atualização dos pacotes.

18.2.1 pkgtool

pkgtool(8) é um programa em formato de menu que permite instalar e remover pacotes. O menu principal é exibido na figura abaixo.

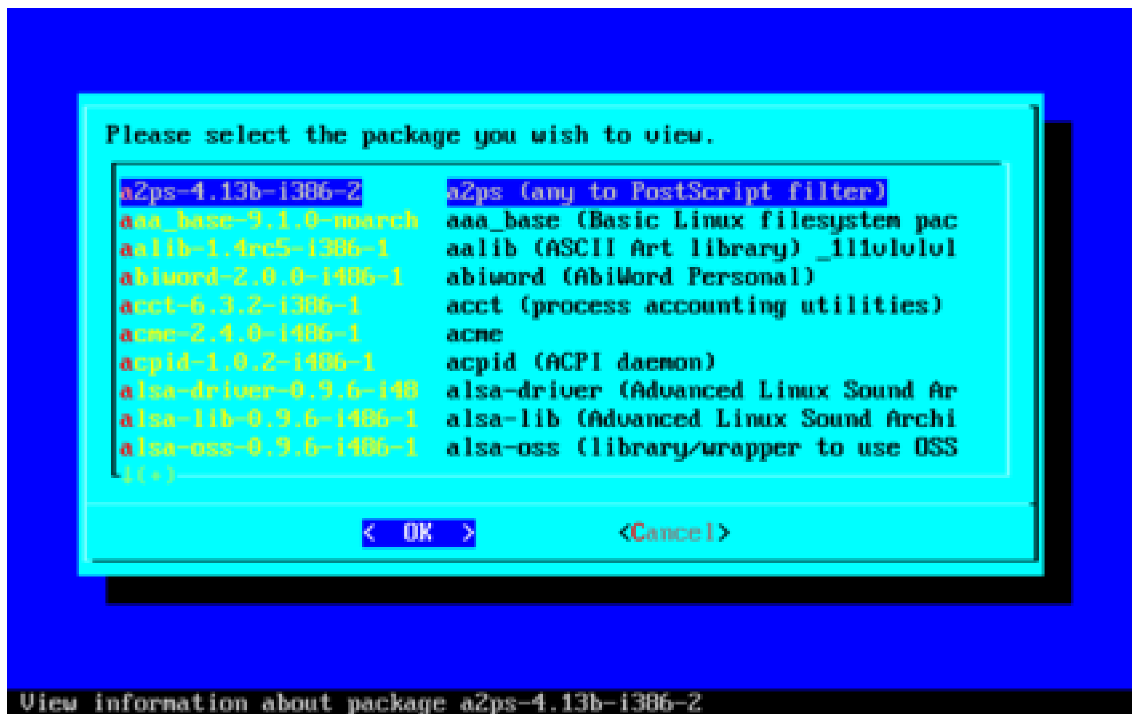
Menu Principal do Pkgtool



A instalação é realizada a partir do diretório atual, outro diretório, ou a partir de disquetes. Basta simplesmente selecionar o tipo de instalação que o pkgtool irá procurar pacotes válidos para serem instalados no local selecionado.

Você também pode visualizar uma lista de todos os pacotes instalados como é mostrado na figura abaixo.

Modo de visualização Pkgtool



Caso você desejar remover algum pacote, escolha a opção remover e então serão apresentados todos os pacotes instalados cada um com um checklist. Selecione os pacotes que deseja remover e então selecione OK. O *pkgtool* removerá os pacotes.

Alguns usuários preferem o utilitário em linha de comando. Entretanto, deve-se notar que cada linha de comando oferece muito mais opções. Além disso, a capacidade para atualizar pacotes somente é possível através de utilitários de linha de comando.

18.2.2 installpkg

O *installpkg*(8) permite instalar um novo pacote no sistema. A sintaxe é mostrada abaixo:

```
# installpkg opções nome_pacote
```

Três opções podem ser utilizadas com o *installpkg*. Porém somente uma opção pode ser utilizada por vez.

Opções do *installpkg*

Opção	**Descrição**
-m	Executa o makepkg no diretório atual.
-warn	Mostra o que aconteceria se você instalar o pacote especificado. Isto é útil para sistemas em pr
-r	Instala todos os pacotes recursivamente a partir do diretório atual. Pode-se utilizar coringas n

Caso você especifique antes a variável de ambiente *ROOT installpkg*, utilizará o caminho especificado. Isso é útil na configuração de novos drives para o seu diretório raiz. Eles serão montados tipicamente em */mnt* ou em algum outro lugar diferente de */*.

O banco de dados com os pacotes instalados é armazenado em */var/log/packages*. O registro é um arquivo texto, sendo um para cada pacote. Se o pacote possui um script de pós-instalação, este é escrito em */var/log/scripts/*.

Você também pode especificar multiplos pacotes ou utilizar coringas no nome do pacote. O *installpkg* não o avisará em caso de reinstalação do pacote existente. Simplesmente instalará o novo pacote sobre o pacote antigo. Caso você queira remover com toda a segurança os arquivos antigos do pacote enterior você deverá utilizar o comando *upgradepkg*.

18.2.3 removepkg

O *removepkg*(8) permite remover os pacotes instalados no sistema. A sintaxe é mostrada abaixo:

```
# removepkg option package_name
```

Quatro opções podem ser utilizadas com o *removepkg*. Porém somente uma opção pode ser utilizada por vez.

Opções do *removepkg*

Opção	**Ação**
-copy	É criado uma árvore de diretórios do pacote preservando o pacote original enquanto é removi
-keep	São criados arquivos temporários durante a remoção. Possuem a finalidade de debug.
-preserve	O pacote é removido, mas uma cópia é realizada para preservar o pacote durante a remoção.
-warn	Exibe informações sobre a remoção do pacote.

Se você passar a variável de ambiente $ROOT$ antes do *removepkg*, tal caminho será utilizado como diretório raiz. Isso é útil na configurarção de novos drives para o seu diretório raiz. Eles serão montados tipicamente em */mnt* ou em algum outro lugar diferente de */*.

O *removepkg* procura os pacotes instalados e somente remove os arquivos referentes ao pacote que você especificou. Também irá procurar por um script de pós-instalação para o pacote especificado e removerá os links simbólicos que foram criados pelo pacote.

Durante o processo de remoção, um status é mostrado. Após a remoção, o registro no banco de dados de pacote é movido para o arquivo */var/log/removed_pacotes* e o script de pós-instalação é movido para */var/log/removed_*scripts.

Assim como o *installpkg*, você pode espeficicar os pacotes pelo nome ou utilizando os coringas para o nome do pacote.

18.2.4 upgradepkg

O *upgradepkg*(8) atualiza um pacote instalado do Slackware package. A sintaxe é mostrada abaixo:

```
# upgradepkg nome_do_pacote
```

ou

```
# upgradepkg nome_pacote_antigo %nome_novo_pacote
```

O *upgradepkg* primeiro instala o novo pacote e em seguida remove o pacote antigo. Assim os arquivos antigos não ficarão no sistema. Caso o nome do pacote tenha sido alterado, utilize a sintaxe do sinal de porcentagem para informar o pacote antigo (aquele que está instalado) e o novo pacote (aquele que você está atualizando). Se você passar a variável de ambiente $ROOT$ antes do *upgradepkg*, tal caminho será utilizado como diretório raiz. Isso é útil na configuração de novos drives para o seu diretório raiz. Eles serão montados tipicamente em */mnt* ou em algum outro lugar diferente de */*.

O *upgradepkg* não é livre de falhas. Você sempre deverá realizar um backup dos seus arquivos de configuração. Caso eles sejam removidos ou substituídos, você terá uma cópia dos arquivos originais caso necessite de algum reparo no sistema.

Assim como o *installpkg* e *removepkg*, você pode espeficicar os pacotes pelo nome ou utilizando os coringas para o nome do pacote.

18.2.5 rpm2tgz/rpm2targz

Atualmente o Gerenciador de Pacotes da Red Hat é o sistema mais popular. Muitos softwares são distribuídos no formato RPM. Mas este formato não é o nosso padrão, nós não recomendamos que as pessoas confiem neles. Entretanto, algumas coisas só estão disponíveis em pacotes RPM (até mesmo o código fonte).

Nós temos um programa que converte pacotes RPM em nosso formato padrão *.tgz*. Isto é realizado, extraíndo o pacote (talvez com o *explodepkg*) para um diretório temporário e examinando o seu conteúdo.

rpm2tgz é o programa utilizado para criar um pacote Slackware com a extensão *.tgz* enquanto que *rpm2targz* cria um arquivo com a extensão *.tar.gz*.

18.3 Criando Pacotes

Criar um pacote para o Slackware pode ser uma tarefa fácil ou difícil. Não será especificado um método para a construção de um pacote. O único requisito para um pacote é ser um arquivo gzipado e, se houver algum script de pós-instalação, este deve estar no arquivo */install/doinst.sh*.

Se você é interessado em criar pacotes para o seu sistema ou para a rede que você gerencia, você deverá dar uma olhada nos diversos build scripts na árvore fonte do Slackware. Existem diversos métodos que podem ser utilizados para criar pacotes.

18.3.1 explodepkg

O *explodepkg*(8) faz a mesma coisa que o *installpkg* para extrair os arquivos do pacote, porém não instala e não registra o pacote no banco de dados. Apenas extrai os arquivos no diretório atual.

Se você verificar a árvore de fontes do Slackware, você verá que pode-se utilizar o comando para *alicerçar* os pacotes. Estes pacotes possuem um esqueleto que criará o pacote final. Ele contém todos os nomes de arquivos (tamanho-zero), permissões, e donos. O build script criará um pacote a partir do diretório fonte do pacote .

18.3.2 makepkg

O *makepkg*(8) cria um pacote válido para o Slackware a partir do diretório atual. Ele buscará na árvore de diretório por links simbólicos, adicionará e criará um script de pós-instalação durante a criação do pacote. Avisará por arquivos de tamanho zero na sua árvore de pacotes.

Este comando é usado após você criar sua árvore de pacotes.

18.3.3 SlackBuild Scripts

Pacotes Slackware são criados de muitas maneiras diferentes de acordo com a necessidade. Nem todos os pacotes são criados pelos próprios programadores que os criaram. Muitos compilam com opções que não são inclusas nos pacotes do Slackware. Caso necessite de alguma funcionalidade da qual o pacote não foi compilado, você deverá criar seu próprio pacote. Felizmente para muitos pacotes do Slackware, você pode encontrar scripts SlackBuild no código fonte do pacote.

Mas o que é um script SlackBuild? Scripts SlackBuild são shell scripts executáveis que você executa como *root* para configurar, compilar, e criar pacotes Slackware. Você pode modificar os scripts e criar seus pacotes Slackwares com sua necessidades específicas.

18.4 Criando Tags e Tagfiles (para configuração)

O programa de configuração do Slackware gerencia a instalação dos pacotes no seu sistema. Existem arquivos que dizem ao programa de configuração quais pacotes devem ser instalados, quais são opcionais, e quais são selecionados por padrão pelo programa de configuração.

Um tagfile está no diretório do primeiro conjunto de software e é chamado de tagfile. Ele lista os pacotes daquele disco em particular e o status, que pode ser:

Status das Opções do Tagfile

Opção	Característica
ADD	O pacote é necessário para a execução necessária do sistema
SKP	O pacote será automaticamente pulado
REC	O pacote não é necessário, mas recomendado
OPT	O pacote é opcional

O formato é simples:

```
nome_pacote: status
```

Um pacote por linha. As tagfiles originais de cada pacote são armazenadas como um tagfile.org. Dessa forma, caso bagunce o seu, você pode restaurar o original.

Muitos administradores preferem escrever seus próprios tagfiles e recomeçar o instalador selecionando *full*. O programa da instalação irá ler os tagfiles e executará a instalação de acordo com seus índices. Caso você utilize REC ou OPT, uma caixa de diálogo será apresentada ao usuário perguntando se quer ou não um pacote específico. Recomenda-se que você utilize ADD e SKP ao escrever tagfiles para automatizar instalação.

Certifique-se apenas que seus tagfiles foram gravados no mesmo lugar que os originais. Ou você pode especificar um tagfile customizado, criado de acordo com as suas necessidades.

Capítulo 19

ZipSlack

19.1 O que é o ZipSlack?

O ZipSlack é uma versão especial do Slackware Linux. Ele é uma versão já instalada do Slackware que está pronta para rodar a partir da sua partição DOS ou Windows. É uma instalação básica, e não tem todos os recursos do Slackware.

O ZipSlack tem esse nome porque é distribuído em um grande arquivo .ZIP. Usuários de DOS e Windows provavelmente terão mais familiaridade com esse tipo de arquivo. Eles são arquivos compactados. O arquivo do ZipSlack contém tudo que você precisa para utilizar o Slackware.

É importante notar que o ZipSlack é significativamente diferente de uma instalação normal. Apesar de funcionarem da mesma maneira e terem os mesmos programas, o público alvo e a suas funções são diferentes. As principais vantagens e desvantagens do ZipSlack são discutidas abaixo.

Uma última dica, você deve sempre consultar a documentação incluída no diretório do ZipSlack. Ela contém as últimas informações relativas a instalação, boot e uso do produto.

19.1.1 Vantagens

- Não requer o reparticionamento do seu disco rigido.
- Uma ótima maneira de se aprender a utilizar o Slackware sem ter que passar pelo processo de instalação.

19.1.2 Desvantagens

- Usa o sistema de arquivos do DOS, que é mais lento que um sistema de arquivos nativo do Linux.
- Não funciona com o Windows NT.

19.2 Obtendo o ZipSlack

Obter o ZipSlack é fácil. Se você comprou a caixa de CDs oficial do Slackware, então você já tem o ZipSlack. Apenas encontre o CD que contém o diretório *zipslack* e coloque-o no seu drive de CD-ROM. Esse CD costuma ser o terceiro ou o quarto disco, mas sempre confie mais nas etiquetas do que nessa documentação já que o disco em que ele está pode mudar.

Se você quiser fazer o download do ZipSlack, você deve primeiro vistar nossa página *Get Slack* para obter as últimas informações referentes a downloads:

http://www.slackware.com/getslack

O ZipSlack faz parte de todas as versões do Slackware. Localize a versão que você quer e vá para o diretório da mesma no FTP. A última versão pode ser encontrada no seguinte endereço:

ftp://ftp.slackware.com/pub/slackware/slackware/

Você encontrará o ZipSlack no subdiretório */zipslack*. Ele é distribuído como um grande arquivo *.ZIP* ou em partes do tamanho de um disquete. As partes estão no diretório */zipslack/split*.

Não fique apenas com os arquivos *.ZIP*. Você também pode fazer o download da documentação e qualquer imagem de boot que estiver no diretório do ZipSlack.

19.2.1 Instalação

Após ter feito o download dos componentes necessários, você precisará extrair o arquivo *.ZIP*. Assegure-se que está usando um descompactador 32-bit. O tamanho e os nomes dos arquivos são muito grandes para um descompactador de 16-bit. Exemplos de descompactadores 32-bit são o WinZip e o PKZIP para Windows.

O ZipSlack é desenvolvido para ser descompactado no diretório raiz de uma unidade (tal como *C:* ou *D:*). Um diretório *LINUX* onde estará a instalação do Slackware será criado. Você também encontrará os arquivos necessários para iniciar (bootar) o sistema nesse diretório.

Após descompactar os arquivos, você terá um diretório *LINUX* na unidade que você escolheu (nós usaremos *C:* a partir de agora).

19.3 Bootando o ZipSlack

Há varias maneiras de se bootar o ZipSlack. A mais comum é usar o arquivo LINUX.BAT para bootar o sistema a partir do DOS (ou a partir do modo DOS do Windows 9x). Esse arquivo deve ser editado para refletir seu sistema antes de utilizá-lo.

Comece abrindo o arquivo //C:\LINUX\LINUX.BAT // em seu editor de textos favorito. No topo do arquivo você verá um grande comentário. Ele explica o que você precisa editar nesse arquivo (e também o que fazer se você estiver bootando atravé de um Zip drive externo). Não se preocupe se você não entender o paramêtro <literal>root=</literal>. Há vários exemplos, sinta-se a vontade para escolher um e testá-lo. Se não funcionar, você pode editar o arquivo novamente, comentar a linha que você usou, e escolher outra.

Após habilitar a linha que você quiser, removendo a palavra *rem* do começo da linha, salve o arquivo e saia do editor de textos. Reboot sua máquina em modo DOS.

Um prompt do DOS no Windows 9x NÃO é suficiente.

Digite C:\LINUX\LINUX.BAT para bootar o sistema.

Se tudo correr bem, você deverá ver um prompt de login.

Logue como *root*, sem senha. Você provavelmente vai querer definir uma senha para root, assim como adicionar uma conta para você. A partir desse ponto você pode consultar outras seções desse livro sobre a utilização do sistema.

Se você não conseguir bootar o sistema usando o arquivo *LINUX.BAT*, você deverá consultar o arquivo *C:\LINUX\README.1ST* incluso sobre outras formas de boot.

Capítulo 20

Glossário

Ambiente gráfico

Uma interface gráfica para o usuário (GUI - Graphical User Interface) que é executado sobre o sistema de janelas X e provê as funcionalidades dos aplicativos instalados, interface coerente entre programas e componentes, gerenciadores de janelas e arquivos, etc. Um passo a frente de um simples gerenciador de janelas.

Área de troca (Swap)

Espaço do disco rígido que é utilizado pelo kernel como uma memória RAM *virtual*. É mais lento que a memória RAM, mas como espaço em disco é mais barato, a área de troca (swap) pode ser mais abundante. A área de troca é útil para o kernel manter dados não muito utilizados e como opção quando a memória RAM está sendo toda utilizada.

Arquivo oculto (Dot file)

No Linux, os nomes de arquivos ocultos começam com um ponto ('.').

Biblioteca

Uma coleção de funções que podem ser compartilhadas entre programas.

Carregador dinâmico

Quando programas são compilados no Linux, eles geralmente usam pedaços de código (funções) de bibliotecas externas. Quando estes programas executam, estas bibliotecas devem ser encontradas e as funções requeridas devem ser carregadas na memória. Esta é a função do carregador dinâmico.

Código fonte

O código (mais ou menos) legível aos humanos com o qual a maioria dos programas são escritos. Códigos fontes são compilados em código *binário*.

Comandos internos do interpretador de comandos

Um comando do interpretador de comandos, ao invés de ser fornecido por um programa externo. Por exemplo, o *bash* tem o comando interno *cd*.

Compilar

Converter um código fonte em código binário.

Conjunto de ferramentas GUI

Um conjunto de ferramentas GUI, é uma coleção de bibliotecas que permite ao programador, a partir de códigos, desenhar *componentes gráficos* como barras de rolagem, checkboxes, etc. e construir uma interface gráfica. O conjunto de ferramentas GUI usado por um programa, geralmente define seu *aspecto*.

Conta

Todas informações de um usuário, incluindo login, senha, informações para o finger, UID, GID e diretório base. Criar uma conta é adicionar um usuário e criar suas definições.

Daemon

Um programa designado para exectar em segundo plano e, sem a intervenção de usuários, desempenha uma tarefa específica (geralmente provendo um serviço).

Darkstar

O hostname padrão do Slackware; seu computador terá o nome de darkstar se você não especificar outro. Um dos computadores de desenvolvimento de Patrick Volkerding, nomeado depois de *Dark Star*, uma música de Grateful Dead

Diretório base

O diretório base *home* de um usuário é o diretório em que o usuário é colocado imediatamente após registrar-se no sistema. Os usuários tem permissões totais é, mais ou menos, um reino livre em seu diretório base.

Diretório raíz

Representado por /, o diretório raíz existe no topo do sistema de arquivos, com todos os outros diretórios se ramificando abaixo dele em uma *árvore de arquivos.*

Diretório de trabalho

O diretório em que um programa se considera estar enquanto está executando.

Disco de inicialização

Um disquete contendo um sistema operacional (no nosso caso, o kernel do Linux) com o qual é possível iniciar um computador.

Disco raíz (rootdisk)

O disco (geralmente fixo) no qual o diretório raíz está armazenado.

Disco suplementar

No Slackware, é um disquete usado durante a instalação que não contém nem o kernel (que está no disco de inicialização), nem o sistema de arquivos raiz (que está no disco raiz), mas contem arquivos adicionais necessários como módulos para rede ou suporte para PCMCIA.

DNS

Domain Name Service (Sistema de Nomes de Domínios). Um sistema que traduz nomes para endereços numéricos em uma rede de computadores.

Dotted quad

Formato do endereço IP, é assim chamado pois consiste em quatro números (na faixa decimal 0-255) separados por pontos.

Drivers de dispositivos

Um pedaço de código no kernel que controla diretamente uma parte do hardware.

Entrada padrão (stdin)

A entrada de dados padrão do Unix. Dados podem ser redirecionados ou concatenados para a stdin dos programas, vindos de qualquer fonte.

Época

Um período da história; em Unix, *a Época* começa às 00:00:00 UTC Janeiro 1, 1970. Esta é considerada a *alvorada do tempo* pelo Unix e sistemas operacionais Unix-like, e todas outras datas são calculadas em relação a esta.

Framebuffer

Um tipo de dispositivo gráfico; no Linux, geralmente se refere ao programa framebuffer, o qual provê uma interface framebuffer padrão para as aplicações enquanto mantém drivers específicos ocultos. Esta camada de abstração libera os programas de *falarem* com vários drivers de dispositivos.

FTP

O protocolo de transferência de arquivos (File Transfer Protocol). FTP é um método bastante utilizado para transferir dados entre computadores.

Gateway

Um computador que repassa dados de uma rede para outra.

Gerenciador de janelas

Um programa gráfico (ambiente X) que provê uma interface gráfica além do simples desenho retangular do Sistema de janelas X. Gerenciadores de janelas geralmente provêem barras de título, menus para os programas em execução, etc.

GID

Identificador de grupo (Group IDentifier). O GID é um número único atribuído a um grupo de usuários.

Grupo

Usuários em Unix pertencem a *grupos*, que podem conter vários usuários e são utilizados para controles de acessos mais generalizados, que é bem mais prático que dar permissões para cada usuário individualmente.

GUI

Interface gráfica do usuário (Graphical User Interface). Uma interface de programas que utiliza elementos gráficos como botões, barras de rolagem, janelas, etc. ao invés de somente entrada e saída de textos.

HOWTO

Um documento descrevendo *como fazer (how to)* algo, como configurar um firewall ou gerenciar usuários e grupos. Existe uma vasta coleção deste tipo de documento disponível no projeto de documentação Linux (LDP - Linux Documentation Project).

HTTP

O protocolo de transferência de hiper-texto (HyperText Transfer Protocol). HTTP é o protocolo primário, no qual, a World Wide Web (WWW) opera.

ICMP

Protocolo de controle de mensagens na Internet (Internet Control Message Protocol). Um protocolo muito básico, utilizado geralmente para pings.

Interface de Rede

Uma representação virtual de um dispositivo de rede, provido pelo kernel. Interfaces de rede permitem usuários e programas comunicarem com dispositivos de rede.

Interpretador de comandos (shell)

Os interpretadores de comandos provêem uma interface de linha de comando para o usuário. Quando você estiver olhando para uma tela em modo texto, você está em um interpretador de comandos.

LILO

O carregador do Linux (LInux LOader). O LILO é o gerenciador de inicialização (boot) mais utilizado com o Linux.

Link simbólico

Um arquivo especial que simplesmente aponta para a localidade de um outro arquivo. Links simbólicos são usados para evitar duplicação de dados quando é necessário ter um determinado arquivo em vários locais.

LOADLIN

LOADLIN é um programa que é executado no MS-DOS ou Windows e inicializa um sistema Linux. É geralmente usado em computadores com múltiplos sistemas operacionais (incluindo Linux e DOS/Windows, é claro).

MBR

O registro mestre de inicialização (Master Boot Record). Um espaço reservado no disco rígido onde estão armazenadas as informações sobre o que fazer durante a inicialização. LILO e outros gerenciadores de inicialização podem ser gravados aqui.

Módulo do kernel

Um pedaço do código do kernel, geralmente um tipo de driver, que pode ser carregado ou descarregado da memória separadamente do corpo principal do kernel. Módulos são úteis para atualizar drivers ou testar configurações do kernel, porque eles podem ser carregados e descarregados sem reiniciar.

MOTD

Mensagem do dia (Message Of The Day). O motd (armazenado no Linux em */etc/motd*) é um arquivo de texto que é exibido a todos usuários durante seu registro no sistema (login). Tradicionalmente, é utilizado pelo administrador de sistemas como um *quadro de avisos* para se comunicar com os usuários.

Motif
Um conjunto de ferramentas de programação popularmente utilizado em muitos programas X antigos.

NFS
O sistema de arquivo de rede (Network FileSystem). NFS permite montar sistemas de arquivos remotos como se fossem locais, provendo assim, um método transparente de compartilhamento de arquivos.

Nó de Dispositivo (Device Node)
Um tipo de arquivo especial no sistema de arquivos */dev* que representa um componente de hardware para o sistema operacional.

Nome de domínio
O nome DNS de um computador, excluindo seu hostname.

Núcleo (Kernel)
O coração do sistema operacional. O núcleo é a parte que provê controle básico de processos e interfaces com o hardware do computador.

Octal
Sistema de números de Base-8, com dígitos de 0-7.

Pacote de programas
Um programa e os arquivos a ele associados, empacotados e comprimidos em um único arquivo, junto com os scripts ou informações necessárias para ajudar no gerenciamento da instalação, atualização e remoção destes arquivos.

Pacote de senhas Shadow
O pacote de senhas shadow permite que senhas criptografadas sejam escondidas dos usuários, enquanto o resto das informações do arquivo */etc/passwd* permanecem visíveis para todos. Isto ajuda a prevenir ataques de *brute-force* para quebrar senhas.

Pager
Um programa gráfico (X) que permite aos usuários visualisar e alternar entre várias *áreas de trabalho*.

Partição
Uma divisão do disco rígido. Sistemas de arquivos residem sobre as partições.

Ponto de montagem
Um diretório vazio em um sistema de arquivos onde um outro sistema de arquivos pode ser *montado* ou anexado.

PPP
Protocolo Ponto-a-Ponto (Point-to-Point Protocol). PPP é utilizado principalmente para conectar, via modem, a um provedor de serviços de Internet (ISP - Internet Service Provider).

Primeiro plano
Um programa que aceita entrada de um terminal é dito que está sendo executado em primeiro plano.

Processo
Um programa em execução.

Processo suspenso
Um processo que foi congelado até que seja morto ou reiniciado.

Programa wrapper
Um programa cuja a única finalidade é executar outros programas, mas que as vezes muda seu comportamento alterando seus ambientes ou filtrando sua entrada.

Runlevel
O estado geral do sistema, definido pelo init. Runlevel 6 significa reíniciar o computador (reboot), runlevel 1 é o *modo monousuário*, runlevel 4 é o login em modo gráfico, etc. Existem 6 runlevels disponíveis em um sistema Slackware.

Saída padrão (stdout)

A saída padrão de informações no Unix. Textos normais de saída são reportados na stdout, que é separada das mensagens de erro, reportadas na stderr, e podem ser concatenadas ou redirecionadas para a stdin de outros programas ou para um arquivo.

Saída padrão de erro (stderr)

O padrão Unix para reportar erros. Os programas reportam qualquer mensagem de erro na stderr, desta forma esta pode ser separa de uma saída normal.

Seção Man

Páginas no padrão de manuais online do Unix (man) são agrupadas em seções para facilitar referências. Todas as páginas sobre programação em C estão na seção 3, páginas sobre administração de sistemas estão na seção 5, etc.

Segundo plano (Background)

Qualquer processo que esteja executando e não aceite entradas ou controles do terminal é dito que está sendo executado em segundo plano.

Séries de programas

Uma coleção de pacotes de softwares relacionados no Slackware. Todos os pacotes do KDE estão na série *kde*, pacotes de rede estão na série *n*, etc.

Serviço

O compartilhamento de informações e/ou dados entre programas ou computadores de um único *servidor* para múltiplos *clientes*. HTTP, FTP, NFS, etc. são serviços.

Servidor de nomes

Um servidor de informações DNS. Servidores de nomes traduzem nomes DNS em endereços IP numéricos.

Servidor X

Um programa no Sistema de Janelas X que faz a interface com o hardware gráfico e abriga os programas gráficos que estão em execução.

Shell seguro (SSH - Secure SHell)

Um método encriptado (assim seguro) de se conectar remotamente a um computador. Muitos programas de shell seguro estão disponíveis; ambos cliente e servidor são necessários.

Sinal

Programas Unix podem comunicar uns com os outros através de simples *sinais*, os quais são enumerados e geralmente possuem significados específicos. *kill -l* listará os sinais disponíveis.

Sistema de arquivos

Uma representação dos dados armazenados onde *arquivos* de dados são mantidos organizados em *diretórios*. O sistema de arquivos é a forma de representação, mais próxima da universal, de dados armazenados em discos (tanto fixos como removíveis).

Sistema de janelas X

Interface gráfica com suporte a rede usada na maioria dos sistemas operacionais *Unix-like*, incluindo o Linux.

SLIP

Protocolo de interface de linha serial (Serial Line Interface Protocol). SLIP é um protocolo similar ao PPP e é utilizado para conectar duas máquinas via interface serial.

Sub-rede

Uma faixa de endereços IP que é parte de uma faixa maior. Por exemplo, 192.168.1.0 é uma sub-rede de 192.168.0.0 (onde 0 é uma mascara *indefinida*); é, na verdade, a sub-rede *.1*.

Superbloco

No Linux as partições são definidas em blocos. Um bloco possui 512 bytes. O superbloco são os primeiros 512 bytes de uma partição.

Tabela de roteamento

Um conjunto de informações que o kernel utiliza para *rotear* dados ao redor da rede. Possui informações como: onde está o seu gateway default, qual interface de rede está conectada a determinada rede, etc.

Tagfile

Um arquivo usado pelo programa de configuração (*setup*) do Slackware durante a instalação, o qual descreve um conjunto de pacotes a serem instalados.

Terminal

Uma interface homem-máquina que consiste de pelo menos uma tela (ou tela virtual) e de algum método de entrada de dados (geralmente um teclado).

Terminal virtual

O uso de programas para simular múltiplos terminais enquanto utiliza um simples conjunto de dispositivos de entrada/saída (teclado, monitor, mouse). Um conjunto especial de teclas, ao serem pressionadas, alterna entre os terminais virtuais a partir de um único terminal físico.

UID

Identificador de usuário (User IDentifier). Um número único que identifica um usuário no sistema. UIDs são usados pela maioria dos programas ao invés dos nomes de usuário, uma vez que números são mais fáceis de manejar; nomes de usuário geralmente são usados somente quando o usuário precisa ver as coisas acontecendo.

Variável de ambiente

Uma variável configurada no shell do usuário que pode ser referenciada pelo usuário ou por programas executados por este usuário nesta shell. Variáveis de ambiente são geralmente utilizadas para armazenar preferências e parâmetros padrão.

VESA

Associação de padrões eletrônicos de video (Video Electronics Standards Association). O termo *VESA* é geralmente usado para denotar um padrão especificado por esta associação. Atualmente todos os adaptadores de vídeo modernos são compatíveis com VESA.

Capítulo 21

The GNU General Public License

GNU GENERAL PUBLIC LICENSE

21.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

21.2 TERMS AND CONDITIONS

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

- This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The ”Program”, below, refers to any such program or work, and a ”work based on the Program”means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term ”modification”.) Each licensee is addressed as ”you”.

  Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

- You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

  You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

- You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

- You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

- If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

- You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
  - Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
  - Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
  - Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

- You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

  - You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

  - Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

  - If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License

and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

- If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

- The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and ”any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

- If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

- NO WARRANTY

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM ”AS IS”WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

- IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

21.3 How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright"line and a pointer to where the full notice is found.

//<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author> //

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA//

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

"' *Gnomovision version 69, Copyright (C) year name of author*

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details. "'

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items–whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer"for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Capítulo 22

Contributing Authors

Author	Contribution
David Cantrell	Co-author of first edition.
Alan Hicks	Project Organizer, lead author of second edition.
Logan Johnson	Co-author of first edition.
Keith Keller	Wrote the wireless networking section.
Joost Kremers	Contributed the Emacs chapter.
Chris Lumens	Co-author of first edition.
Jurgen Philippaerts	Updated the Basic Networking Commands chapter.
David G. Sizemore	Updated the Introduction
Murray Stokely	Wrote the Preface, index, and prepared the book for print publication
Simon Williams	Contributed the Emacs chapter.

22.1 Contribuidores para o Português do Brasil

Diogo Leal (estranho)	Tradução	Revisão	Site
Bruno T. Russo (BrunoRusso)	Tradução	Revisão	Site
Ellington Santos	Revisão	Repositório SVN	
Artur Souza (MoRpHeUz)	Tradução	Revisão	
Thiago Diniz	Tradução	Revisão	
José Luiz Moquenco de Figueiredo (zecafig)	Tradução	Revisão	
Waldemar Silva Júnior (wsjunior)	Tradução	Revisão	
João Carlos de Oliveira Coutinho (BitDesigner)	Tradução	Revisão	
Raphael Bastos (ChemonZ)	Compilação		
Gabriel Favaro	Tradução	Revisão	
Rodrigo Amorim Ferreira	Tradução	Revisão	
Tiago T3	Tradução	Revisão	
Neumar Malheiros	Tradução	Revisão	
Gabriel Marrocos Magalhães (Manda_Chuva)			
Ariel			
Jairo			
Lucas			
Frederico			

Agradecemos muito a todos que contribuiram, fazendo com que este projeto fosse possível